DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA MOTOCICLETAS (SISMO)

AUTORES:

JOSÉ JULIÁN RAMOS ROJAS

PEDRO JOSÉ JIMÉNEZ PÉREZ

DIRECTOR DEL PROYECTO:

MAG. PEDRO GUEVARA

TRABAJO DE GRADO

X SEMESTRE

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

UNIVERSIDAD DE CÓRDOBA

SAHAGÚN – CÓRDOBA

2015

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA MOTOCICLETAS (SISMO)

JOSÉ JULIÁN RAMOS ROJAS
PEDRO JOSÉ JIMÉNEZ PÉREZ

TRABAJO DE GRADO PRESENTADO EN CUMPLIMIENTO PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

DIRECTOR DEL PROYECTO:
MAG. PEDRO GUEVARA

TRABAJO DE GRADO
X SEMESTRE

FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
UNIVERSIDAD DE CÓRDOBA
SAHAGÚN – CÓRDOBA
2015

Nota de aceptación

_____

_____

_____

_____

Presidente del jurado

_____

Jurado 1

_____

Jurado 2

Montería, Noviembre de 2015

AGRADECIMIENTOS

Doy gracias a Dios, a mis padres Enrique Ramos y Myriam Rojas, a mis hermanas Sandra, Paola, Kenia y a todos mis familiares que con su apoyo y esfuerzo hicieron esto posible, a la Universidad de Córdoba por permitirme convertir en un profesional y a todos los docentes que hicieron parte de mi proceso integral de formación, a mi compañero de trabajo Pedro Jiménez por sus aportes, dedicación y esfuerzo, a todos mis compañeros de estudio por su compañerismo y colaboración, a mis amigos, por su apoyo moral.

José Julián Ramos Rojas

Doy gracias principalmente a mi madre Gloria Enith Pérez Vega y a mi tía Ana Cristina Flórez Bertel porque sin ellas nada de esto hubiese sido posible, a mis demás familiares por apoyarme en este camino, a la Universidad de Córdoba y a todos los docentes que me guiaron y me brindaron sus conocimientos para poder convertirme en un profesional, a mi compañero José Julián Ramos con quien realice este proyecto y con quien afronte la mayoría de los retos que me impuso esta carrera, a todos mis demás compañeros de estudio por su amistad y por recorrer conmigo este arduo camino y finalmente a mis amigos por estar a mi lado siempre.

A todos les doy gracias por contribuir de una u otra manera en el proceso de convertirme en un Ingeniero de Sistemas.

Pedro José Jiménez Pérez

Contenido

TITULO DEL PROYECTO

LISTA DE FIGURAS

TITULO DEL PROYECTO

Desarrollo e implementación de un sistema de seguridad para motocicletas (SisMo).

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Desarrollar un sistema de seguridad como estrategia para evitar o disminuir las posibilidades de que sea hurtada una motocicleta.

1.2 OBJETIVOS ESPECÍFICOS

- Identificar los componentes de hardware y software adecuados para la construcción del sistema de monitoreo y control.

- Desarrollar una aplicación para el sistema operativo Android capaz de controlar las funcionalidades del dispositivo de monitoreo y control.

- Realizar pruebas que garanticen la funcionalidad del sistema.

2. INTRODUCCIÓN

2.1 AMBIENTACIÓN

El hurto de motocicletas es un fenómeno que ocurre a gran escala en muchos países del mundo, porque este tipo de vehículo es más fácil de hurtar, esconder, “desaparecer” que otros vehículos, además, porque los conductores no implementan ningún control ni medida de seguridad en sus motocicletas facilitándole así el trabajo a las personas que se dedican a esta actividad delictiva.

En Colombia desde el año 2004 hasta el año 2013 fueron hurtadas 143.213 motocicletas, siendo en este periodo el 2013 el año con mayor índice de robos de este tipo de vehículos, con un total de 22.751 casos (El nuevo siglo, 2014). Para el año 2014 se tuvo un reporte de 23.398 motocicletas hurtadas y para el primer trimestre del 2015 van 5.701 denuncias de robo, como señala un informe de la Policía Nacional y Asopartes (El tiempo, 2015). Una buena parte de estas motocicletas terminan en los mercados negros de Venezuela, Ecuador, Perú y de algunos países de Centroamérica. Según informes de la policía nacional, las ciudades que registran los más altos índices en hurto de motocicletas son Medellín, Bogotá y Cali (El nuevo siglo, 2014).

Debido a esta situación surge la necesidad de implementar un sistema de seguridad que permita controlar o por lo menos mitigar las posibilidades de que se roben una motocicleta. Para esto se ha planteado desarrollar un sistema que integre componentes de hardware y software. Estos componentes serán descritos con mayor detalle en el desarrollo de este documento, donde también se abordaran las diferentes etapas necesarias para el diseño, la construcción e implementación de dicho sistema, además, se mostraran proyectos similares que se han implementado en distintos países con el fin de contrarrestar el hurto de motocicletas.

Con la realización de este proyecto se pretende desarrollar un sistema que se encargue de controlar el sistema eléctrico de la motocicleta, monitorear el seguro y la posición de la misma. Luego de dejarla estacionada y que el usuario haya activado el monitoreo, el sistema se encargará de notificar a este a través de su dispositivo móvil cualquier alteración del estado de los elementos monitoreados, con el fin de que el usuario no descuide ni un

segundo su vehículo, aun cuando se encuentre alejado de este. En caso de que el sistema detecte que se ha realizado un posible robo, le brindará información al usuario para facilitar la recuperación de la motocicleta.

2.2 PROBLEMÁTICA

Hurto de motocicletas en Colombia

El alto índice de robo de motocicletas que existe en Colombia, es uno de los muchos problemas que enfrenta el país en la actualidad y que viene creciendo cada vez con más fuerza en muchas regiones de este.

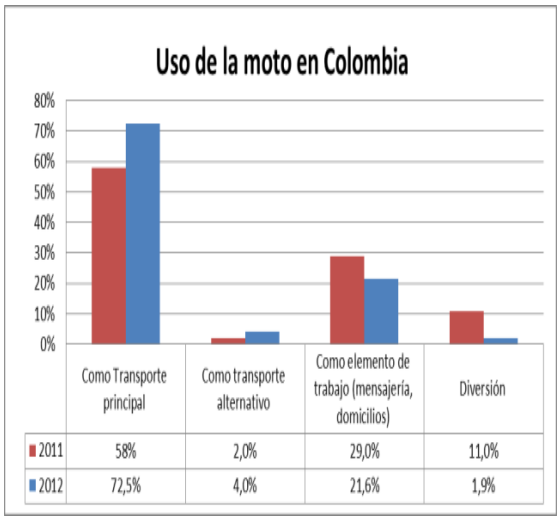

Figura 1: Uso de la motocicleta en Colombia

Analizando el más reciente estudio socio-demográfico del año 2012 elaborado por el Comité de Ensambladoras Japonesas, en Colombia cada vez más se hace evidente la importancia de la motocicleta como herramienta de desarrollo económico y para mejorar la calidad de vida de las personas (Asomocol, 2013). En la siguiente figura se puede observar el comportamiento del uso de la motocicleta en Colombia para los años 2011 y 2012.

Según el Registro Único Nacional de Tránsito, Runt, hasta el 31 de diciembre del 2014, en Colombia había 6’022.451 motocicletas (Motor, 2015). Desde el año 2004 hasta el año 2013 fueron hurtadas 143.213 motocicletas, siendo en este periodo el 2013 el año con mayor índice de robos de este tipo de vehículos, con un total de 22.751 casos (El nuevo siglo,

2014). Para el año 2014 se tuvo un reporte de 23.398 motocicletas hurtadas, pero eso no es todo, el fenómeno criminal parece empeorar este año: la delincuencia no da tregua y en el primer trimestre van 5.701 denuncias de robo, como señala un informe de la Policía Nacional y Asopartes. El año pasado, en la misma época hubo 4.291 casos, lo que representa para el 2015 un aumento en 1.410 reportes dando un promedio diario de 64 motocicletas hurtadas (El tiempo, 2015). Una buena parte de estas motocicletas terminan en los mercados negros de Venezuela, Ecuador, Perú y de algunos países de Centroamérica. Según informes de la policía nacional, las ciudades que registran los más altos índices en hurto de motocicletas son Medellín, Bogotá y Cali (El nuevo siglo, 2014).

El hurto de las motocicletas viene creciendo en Colombia en los últimos meses, pues hay más motocicletas que carros. Además porque las motos son más fáciles de hurtar, esconder y “desaparecer”. Con mucha frecuencia los conductores de estos vehículos las parquean en la vía pública sin ningún control ni medida de seguridad, por esta razón, la modalidad de hurto más usada durante este año por los delincuentes es el 'halado' con (57,7%); seguido por atraco (40%) y un (2,3%) otras modalidades de hurto, como el engaño y la escopolamina. En cuanto a marcas, la lista de las más hurtadas en los primeros tres meses del año la encabeza Yamaha (995), seguida por la Auteco Pulsar (895), Honda (534) y Susuki (442) (El tiempo, 2015).Todos estos datos se pueden observar con mayor claridad en la Fig. 2:

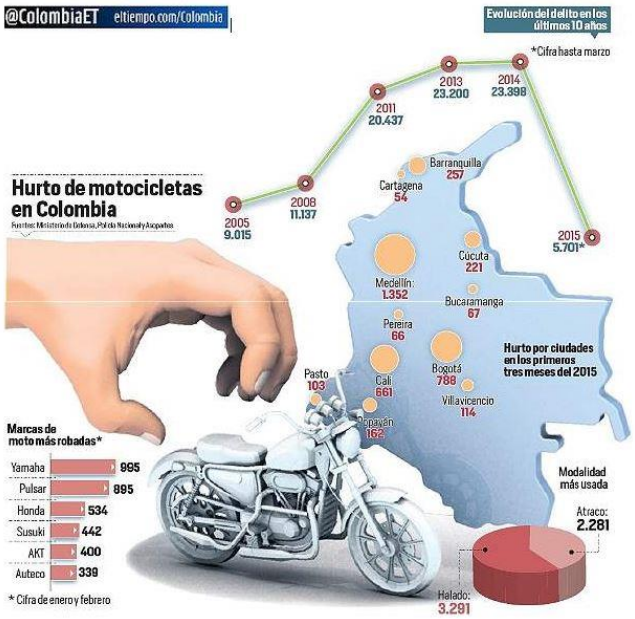

Figura 2: Hurto de motocicletas en Colombia

El alto índice de robo de motocicletas en Colombia se debe, principalmente al descuido de los dueños, también a que algunos negocios se prestan para vender los repuestos de las motocicletas hurtadas a muy bajo costo, además de esto existe poco control y vigilancia en ciertas zonas del país, otra razón de gran peso por la cual se presenta esta problemática es debido a que la legislación colombiana termina siendo muy débil y casi el 99% de los capturados por esta clase de hechos delictivos quedan en libertad (El nuevo siglo, 2014).

La investigación realizada sobre el robo de motos en Colombia, deja visto la gran problemática que se está viviendo en muchas regiones del país en especial en las principales ciudades como Bogotá, Medellín y Cali.

Todo esto conlleva a una gran problemática social y económica por todo lo que una motocicleta representa para el desarrollo personal de un individuo así como para el desarrollo de una región, y que no muestra hasta el momento una tasa de disminución, sino todo lo contrario, sigue una tendencia a la alza de esta problemática. Debido a esto surge la necesidad de implementar en conjunto planes de acciones y herramientas de seguridad que permitan controlar o por lo menos mitigar las posibilidades de que se roben una motocicleta, con el fin de ofrecer mayor seguridad y confianza a los conductores al momento de salir en sus motos.

2.3 ANTECEDENTES

2.3.1 ANTECEDENTES DE LA PROBLEMÁTICA

El problema del hurto de motocicletas no es un caso aislado para Colombia, sino que también es una problemática que se viene presentando en muchos países del mundo, por esta razón se han implementado en los distintos países que se ven afectado por este flagelo, una serie de planes de acción que permiten contrarrestar o por lo menos dificultar el trabajo para los delincuentes que se dedican a este tipo de delito.

Entre los principales planes de acción que se han implementado a nivel mundial, nacional y regional se encuentran:

En Chile, debido a los grandes índices de robo de motocicletas reportados en el año 2010 y 2011 con unas cifras de 1.395 y 1.490 casos respectivamente, se realizó una alianza de trabajo entre la Subsecretaría de Prevención del Delito, la empresa Pats, la empresa IMoto y la Policía de Investigaciones, esta alianza lanza el programa “No Compres Robado”, que tiene como finalidad desbaratar el mercado de bienes robados en el país, promoción de sistemas de marcados de bienes susceptibles de robo y campañas comunicacionales orientadas a prevenir delitos contra la propiedad, y desincentivar la receptación. Entre los objetivos de este programa se encontraba la marcación de las piezas de las motocicletas con un código alfa numérico que permite vincular el objeto marcado con su propietario, a través de una base de datos, esto permite facilitar la identificación y recuperación de bienes en caso de robo, además de dificultar la reventa de estos bienes, por lo que resulta disuasivo para los delincuentes (Francisco Águila V, 2012).

En Cali, Colombia, para el año 2012 circularon alrededor de 112.909 motocicletas y según la Secretaría de Tránsito se hurtaron en promedio cinco de estos vehículos por día. Con el propósito de desestimular el hurto de motocicletas en Cali, la Policía Metropolitana inició una campaña para marcar estos vehículos como los que hacen las compañías aseguradoras. La meta de la Policía fue que la mitad de las motos quedaran marcadas antes de culminar el año. Con un compresor, marcaban el número de la placa en el chasis, las farolas, el exhosto, las tapas del motor y las direccionales. La marcación de estos vehículos fue totalmente gratuita (Diario ADN, 2012).

En otras ciudades del país como Florida blanca - Santander en el año 2010 y Montería – Córdoba en el año 2013 se realizaron campañas educativas y de reacción, que buscaban crear conciencia ciudadana, que las personas tuvieran seguridad sobre sus vehículos y no los dejaran a la mano de la delincuencia. Estas campañas se realizaron debido a los altos índices de robos de motocicletas presentes en estas ciudades y que la principal razón de estos hurtos fue el descuido de los propietarios, que dejaban sus motos en las calles, sobre todo los fines de semana, brindándole la oportunidad al delincuente (Meridiano de Córdoba, 2013), (Vanguardia, 2010).

En Sahagún, Córdoba, Colombia, en el año 2012 a pesar de los consejos de seguridad, de los pronunciamientos del concejo municipal y de los anuncios de las autoridades de combatir los actos delincuenciales, este municipio siguió siendo blanco de robos continuos

de motocicletas. El problema con el hurto de motocicletas se presentaba más que todo los fines de semana, los delincuentes se las llevaban de los parqueaderos de las Iglesias, del hospital San Juan de Sahagún y de los alrededores de supertiendas Olímpica. La ciudadanía quería más resultados por parte de la policía en aras de contrarrestar el delito en la ciudad, por ello saludaron con beneplácito el anuncio hecho por el gobernador de Córdoba Alejandro Lyons, en el sentido de que el ministerio del interior aprobó un proyecto que buscaba organizar varios CAI móviles y dotar de cámaras de vigilancia a 8 municipios de Córdoba, entre ellos Sahagún, lo que sería de gran ayuda para enfrentar a los delincuentes que se dedican a esta actividad (El universal, 2012).

2.3.2 ANTECEDENTES DE LA APLICACION

A continuación se detallaran las herramientas más usadas para contrarrestar el hurto de motocicletas a nivel internacional, nacional así como también en la región cordobesa.

2.3.2.1 Contexto internacional

Existen diferentes herramientas o sistemas antirrobos de uso común a nivel mundial para la problemática del robo de motocicletas, las más utilizadas son:

Horquillas

Son sistemas antirrobo bastante seguros. Los antirrobos de horquilla tienen forma de "u" y están pensados para bloquear alguna de las dos ruedas de la motocicleta. La manera correcta de utilizarlos es en la rueda delantera, uniendo las horquillas y la llanta, dejándola así bloqueada. Si se pone en la llanta trasera, debe unirse la llanta y un amortiguador, también es utilizada para anclar la moto a un punto fijo. Entre los aspectos negativos está su precio, su tamaño y su peso (Roberto Ruiz, SF).

Cadenas

Las cadenas cuentan con fuertes eslabones que permiten movilidad y se adaptan sin dificultad a los elementos a unir. Normalmente se componen por elementos de acero

cementado, son muy pesadas, y poco prácticas para ser llevadas en la moto. Son idóneas para anclar la moto a un punto fijo o para unir varias motos. Las hay de diferentes longitudes y grosores, y suelen ir recubiertas por algún material que protege la moto de ralladuras (Roberto Ruiz, SF).

Antirrobos de disco

Los antirrobos de disco son muy comunes ya que ocupan poco espacio, son ligeros y son fácilmente transportables en la moto. Su misión es la de morder el disco de freno de la moto, el delantero o el trasero, e impedir así su giro, inmovilizando la motocicleta. Además, según el modelo, también pueden ser utilizados para cerrar cadenas (Roberto Ruiz, SF).

Antirrobos con alarma

Son antirrobos comunes, como de disco o de horquilla, equipados con fuertes alarmas sonoras. Cuando el antirrobo está colocado y con la alarma activada al más mínimo movimiento o manipulación comienzan a sonar estridentemente, alertando a cualquier posible testigo (Roberto Ruiz, SF).

Aparte de los dispositivos ya mencionados, en diferentes partes del mundo se han desarrollado otros dispositivos que cuentan con tecnología más avanzada y que proporcionan más funcionalidades a los usuarios. Lo anterior se describirá a continuación.

Detector OnBike. Sistema de seguridad para motocicletas. Madrid – España.

Es un sistema de seguridad para motocicletas desarrollado por la empresa Detector, entre sus funciones principales se encuentran:

Localización del vehículo a través del Smartphone, esta funcionalidad permite conocer donde se encuentra la motocicleta en cualquier momento a través de una aplicación móvil, con un límite de 30 localizaciones al año. Recuperación en caso de robo, gracias al uso de tecnologías de radiofrecuencia, Detector localiza la señal del dispositivo incluso en garajes, contenedores metálicos o en sótanos bajo tierra, lo que facilita la recuperación de las

motocicletas en caso de robo. Aviso de accidente, en el caso de sufrir un impacto brusco, Detector recibirá una alerta y llamará al cliente para verificar su estado. Si no se puede localizar al cliente por teléfono, se avisará a emergencias para que acuda a la posición donde se encuentra la moto (Motor Pasión Moto, 2013).

ADT BIKE SECURITY SERVICE. Las compañías ADT y Gpsmicrosat han desarrollado un dispositivo que combina la tecnología de localización mediante satélite GPS con la de comunicación GSM/GPRS. Madrid- España (2007)

La compañía especializada en seguridad ADT y Gpsmicrosat, especialista en desarrollos GPS, acaban de crear un dispositivo que, una vez instalado en la motocicleta, permite a todos aquellos usuarios tener su vehículo permanentemente localizable, evitando así posibles robos. Además, este desarrollo, que ha sido bautizado como ADT Bike Security Service, lanza un mensaje de alerta a una central de alarmas, en el caso de que el usuario haya sufrido un accidente. El dispositivo es un aparato del tamaño de un paquete de tabaco, que según Ricard Soler, director de Gpsmicrosat, queda completamente oculto en la motocicleta. Este desarrollo combina la tecnología de localización mediante satélite GPS con la de comunicación GSM/GPRS, la que utilizan la mayoría de los teléfonos móviles actuales. A todo ello se suman una serie de sensores que analizan las vibraciones producidas en la moto, la inclinación de la misma y el movimiento. El precio del ADT Bike Security Service es de 499 euros a lo que hay que sumar una cuota de 16 euros mensuales (Jose Luis Cano, 2007).

Localizador de gps CARGO TRACKER SMS. Argentina (2012)

El localizador en tiempo real Cargo Tracker SMS es utilizado para el seguimiento de mercadería, rastreo de vehículos y actividades de espionaje. Este dispositivo permite recibir toda la información sobre las acciones del objetivo controlado desde la comodidad del celular del usuario e implementa diferentes funciones, como alarma por exceso de velocidad o límite perimetral. Posee una batería interna de 1100 mah que otorga una autonomía de hasta 48hs. Se Puede conectar el GPS Tracker a la batería del vehículo. De esa forma la batería del rastreador de gps se recargará cada vez que el vehículo este encendido. Se puede llamar al dispositivo y a través de un Micrófono Inalámbrico GSM Ambiental escuchar el sonido ambiente del lugar. El Localizador de GPS Cargo Tracker SMS funciona en todas partes del

mundo y con cualquier compañía de celular (Bidcom, 2012).

2.3.2.2 Contexto nacional

Tecnología GPS reduciría robos de motocicletas en Colombia (2012)

La multinacional especializada en soluciones de seguridad, monitoreo y localización Detektor, lanzó productos para flotas de motos. Mediante estos dispositivos, las flotillas de motos de los diferentes sectores (mensajería, restaurantes, servicios, etc.) pueden hacer un mejor seguimiento, control de activos, localización y entrega eficiente de productos, y ubicación de activos robados, que es otro de los grandes problemas que enfrentan estos vehículos. Con estos sistemas de geo-localización es posible hacer un seguimiento de los domicilios vía Internet, por medio de códigos entregados al cliente para que verifique en dónde y por qué ruta viene el pedido. La inseguridad es una de las principales preocupaciones para empresas y motociclistas que trabajan como independientes. Por eso estos dispositivos también les permiten a las compañías y al trabajador independiente proteger sus motos en caso de robo. El usuario reporta a la central, se activa el dispositivo y se genera un operativo con fuerza de reacción, que incluye colaboración con las autoridades (Portafolio, 2012).

"Secure Bike" Proyecto Sistema De Seguridad Para Moto. Universidad Del Valle, Sede Yumbo (2013)

Es un dispositivo electrónico que brinda seguridad a una motocicleta a través de un control inalámbrico con la que se puede encender el vehículo automáticamente y que trabaja con un botón de emergencia, que apaga la moto automáticamente si alguien la prende sin el consentimiento del dueño, incluye un dispositivo de alarma celular, el cual llama automáticamente al celular de usuario avisándole si alguien mueve la motocicleta (Secure Bike, 2013).

GPS ubicación satelital, alarma GSM para motos. Medellín (2012)

Es un sistema de ubicación satelital, que permite saber dónde se encuentra ubicada la moto en todo el territorio nacional. A demás de esto permite apagar la moto, bloquear y desbloquear el encendido, obtener las coordenadas de ubicación de la moto, a través de mensajes de texto, cuenta también con una plataforma web que nos permite ver la ubicación

de la motocicleta (ALARCOM, 2012).

Alarma de alejamiento Thunder. Medellín (2012)

Es un dispositivo que evita que enciendan la motocicleta con un elemento distinto a la llave, y que activa una alarma sonora con el mínimo contacto con la motocicleta. Permite encender y apagar el vehículo a distancia. La alarma trabaja de forma automática, al alejarse la motocicleta aproximadamente 100 metros del control de la alarma, el vehículo se apaga de forma automática, con el fin de evitar el robo de esta (Thunder, 2012).

2.3.2.3 Contexto regional

En cuanto a sistemas de seguridad, en la región se utilizan los dispositivos comunes que se describieron anteriormente en este documento.

2.4 JUSTIFICACIÓN

Además de la problemática planteada, para la justificación de este proyecto se tienen en cuenta dos aspectos importantes. El primero consiste en las distintas campañas y planes de acción que se han implementado en algunas ciudades del país, como medidas para contrarrestar el alto índice de robo de motocicletas que se presenta actualmente en Colombia. Encontrando casos como:

En Cali, en el año 2012, con el propósito de desestimular el hurto de motocicletas, la Policía Metropolitana inició una campaña para marcar gratuitamente estos vehículos. Con un compresor, marcan el número de la placa en el chasis, las farolas, el exhosto, las tapas del motor y los direccionales (Diario ADN, 2012). Una campaña similar se realizó en el año 2013 en la ciudad de Bucaramanga, donde el comandante de la Policía Metropolitana, Brigadier General Saúl Torres Mojica afirmó que: “El objetivo de la campaña es evitar el robo directo de las motos o la comercialización de autopartes”, teniendo en cuenta lo afirmado también por el secretario del Interior de Bucaramanga, René Garzón Martínez a cerca de esta campaña: “Será muy difícil para los hampones que roban motos distribuir sus partes y movilizar los vehículos con tranquilidad” (Vanguardia, 2013).

Por otra parte, en la ciudad de Montería, la Policía lanzó una campaña llamada 'No de papaya con su moto', con el fin de concientizar a los propietarios de estos vehículos de no descuidarlos ni dejarlos parqueados en las calles, que es cuando los ladrones aprovechan para cometer el delito. Durante la campaña también se realizaron operativos para la recuperación de motocicletas robadas, la Policía recuperó en un día 12 motocicletas en el centro de la ciudad, luego de hacer diferentes registros en talleres mecánicos y parqueaderos (Meridiano de Córdoba, 2013).

En aras de contrarrestar el delito en la ciudad de Sahagún, el gobernador de Córdoba Alejandro Lyons, anunció que el Ministerio del Interior aprobó un proyecto que busca organizar varios CAI móviles y dotar de cámaras de vigilancia a 8 municipios de Córdoba, entre ellos Sahagún, lo que sería de gran ayuda para enfrentar a los delincuentes que se dedican al hurto de motocicletas en la ciudad (El universal, 2012).

El segundo aspecto, gira en torno al disparo en la venta de motocicletas en el país, siendo el 2014 el mejor año de la historia en ventas para el mercado de autos y de motos en Colombia, en donde este último no para de crecer. De acuerdo con datos suministrados por el Registro Único Nacional de Tránsito, Runt, se matricularon 659.279 unidades en 2014, frente a las 621.695 de 2013 y las 565.374 del 2012. Si bien en 2014 el aumento (37.584 unidades) no superó al registrado en 2013 (56.321), de seguir esta tendencia se estima que en poco tiempo las motos triplicarán el número de autos particulares y que en 2020 la cantidad de esos aparatos de dos ruedas podrá superar los 10 millones en las calles. En la actualidad, Colombia ocupa el segundo lugar en América Latina, después de Brasil, en producción y consumo de motocicletas, campo en el que juegan duro las marcas japonesas con plantas en Colombia como Fanalca-Honda, Incolmotos-Yamaha, Auteco y Suzuki Motor de Colombia (Motor, 2015).

A partir del primer aspecto mencionado, se puede deducir que las campañas implementadas por el Gobierno han contribuido de una manera u otra a contrarrestar o disminuir el hurto de motocicletas en el país, sin embargo, son medidas que no resuelven en su totalidad esta problemática, además, sumándole a esto lo mencionado en el segundo aspecto, donde queda en evidencia el gran crecimiento en las ventas de motocicletas en los últimos años, conlleva a pensar que todo esto generaría una gran demanda de sistemas de seguridad para

motocicletas. Por lo anterior se ha planteado diseñar, construir e implementar un sistema de seguridad complementario para motocicletas. Este sistema será una herramienta que proporcionara seguridad, confianza y tranquilidad a los usuarios, así como también será una ayuda tanto para el usuario como para la fuerza pública en la recuperación de estos vehículos en caso de robo.

El dispositivo ira dirigido a empresas, instituciones y personas que utilicen una motocicleta como medio de transporte o herramienta de trabajo dentro del territorio nacional.

3. MARCO DE INVESTIGACION

3.1 MARCO TEORICO

3.1.1 Sistemas de seguridad

El concepto de seguridad es demasiado amplio y abarca muchos campos. Entre otros aspectos, hay que pensar en la seguridad personal y en la de objetos de cierto valor.

Partiendo de lo anterior, se puede definir como sistema de seguridad, al conjunto de elementos e instalaciones necesarios para proporcionar a las personas y bienes materiales, protección frente a agresiones, robos, atraco, incendio, etc. (CEil, S.F).

Los sistemas de seguridad no sólo sirven para proteger a las personas y a los bienes materiales, sino que también ahorran tiempo y dinero y en los procesos domésticos e industriales su uso está totalmente generalizado.

Algunos ejemplos de su aplicación son (CEil, S.F):

- Seguridad en la vivienda.
- Seguridad en establecimientos.
- Seguridad centrales nucleares.
- Seguridad activa contra incendios.
- Seguridad en calefacción y cuartos de máquinas.
- Seguridad en los vehículos.

Y muchos otros campos donde los sistemas de seguridad proporcionan aplicaciones específicas.

Composición de un sistema de seguridad.

Un sistema de seguridad simple se compone de ciertas partes básicas: unidad de control, sensores, sistemas de aviso y actuadores.

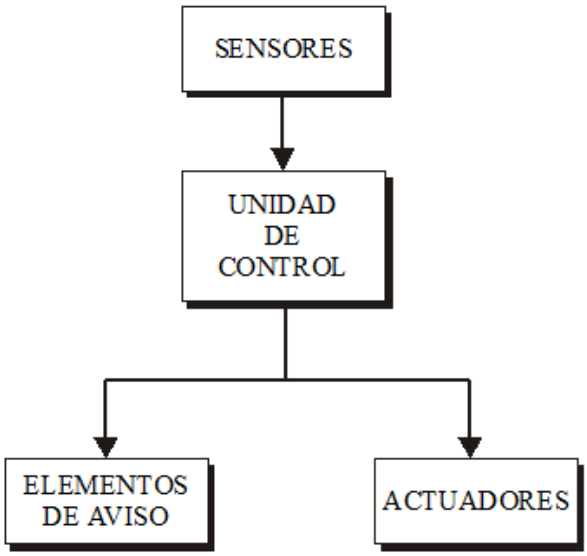

Figura 3: Esquema de composición de un sistema de seguridad

A continuación se realizara una breve descripción de cada uno de los elementos vistos en Figura: 3 (CEil, S.F):

Unidad de control

La unidad de control es la que recibe la señal eléctrica de los detectores o sensores que por algún motivo son activados. Al recibir esta señal, los circuitos electrónicos que lleva en su interior, hacen que se pongan en marcha el sistema de aviso y/o los actuadores.

Sensores

Los sensores son elementos capaces de comprobar las variaciones de una condición de reposo en un lugar determinado y envían información de esa variación a la unidad de control. Son de reducido tamaño y se alimentan a través de una fuente de alimentación de baja tensión.

Sistemas de aviso

Son los dispositivos encargados de avisar de las variaciones detectadas por los sensores dentro del sistema de seguridad. Estos le dan sentido a la instalación de un sistema de seguridad, ya que sin ellos, no tendría sentido la utilización de los sensores y la unidad de control. Pueden ser acústicos (sirenas), ópticos (luces), etc.

Actuadores

Son los dispositivos encargados de ejecutar las órdenes enviadas por la unidad de control en

función de las variaciones detectadas por los sensores dentro del sistema de seguridad. Los actuadores cumplen fundamentalmente dos funciones básicas: prevenir el acceso no autorizado al elemento protegido o minimizar los posibles efectos cuando el acceso ya ha sido vulnerado (Julián Rodríguez Fernández, 2013).

El sistema de seguridad para motocicletas (**SisMo**) cuenta con cada uno de los componentes anteriormente descritos. La unidad de control en este sistema es representada por la placa Arduino Leonardo, la cual se encarga de recibir las señales emitidas por los sensores (Modulo GPS y circuito del seguro) para luego enviar estos datos al sistema de aviso (Aplicación Móvil: encargada de notificarle al usuario cualquier inconveniente con su vehículo) y que a la vez envié una señal al dispositivo actuador del sistema (relé) que se encarga de cortar o permitir el paso de la corriente para el encendido de la motocicleta.

3.1.2 Internet de las cosas

El "Internet de las cosas" (IoT, por sus siglas en inglés), es un concepto que nació en 1999 en el Instituto de Tecnología de Massachusetts (MIT) y que fue definido por primera vez por Kevin Ashton: “El IoT es el mundo en el que cada objeto tiene una identidad virtual propia y capacidad potencial para integrarse e interactuar de manera independiente en la Red con cualquier otro individuo, ya sea una máquina (M2M) o un humano” (Kevin Ashton, 2009). Alternativamente, Internet de las cosas es el punto en el tiempo en el que se conectarían a internet más “cosas u objetos” que personas (Dave Evans, 2011).

IoT es un cambio en la forma de relacionarse los objetos y las personas, incluso los objetos entre sí, donde estos se conectaran entre ellos y con la Red, otorgando datos en tiempo real. Dicho de otra manera, es la digitalización del mundo real.

En los últimos años Cisco System, plantea que estamos en el ‘internet de todo’ o ‘internet of everything’ definiéndolo como: “Fenómeno que reúne a las personas, los procesos, los datos y las cosas para lograr que las conexiones en red sean más relevantes y valiosas que nunca “. En las siguiente grafica se puede observar el crecimiento en el número de conexiones en internet de personas y objetos en los últimos años (El Tiempo, SF):

Figura 4: Crecimiento en el número de conexiones a internet

Según el estudio realizado por Cisco System se calcula que para el año 2020 habrá 4.500 millones de internautas y 50.000 millones de cosas conectadas a internet.

La adopción del internet de las cosas se está acelerando en el mundo debido principalmente a (El Tiempo, SF):

- El crecimiento exponencial de la red que ha generado el interés para el desarrollo de aplicaciones en todos los campos.
- La reducción de tamaños y precios de procesadores y sensores.
- La reducción de costos en almacenaje y procesamiento de grandes volúmenes de información (Big Data).

Para definir de forma correcta la arquitectura y los dispositivos utilizados para este proyecto, se tuvo en cuenta aspectos importantes relacionados con el IoT, entre los que se encuentran (Iván Prieto, 2015):

Dispositivos: Los componentes electrónicos como: procesadores, sensores, baterías, conectores o adaptadores de redes, entre otros, ha alcanzado un elevado grado de optimización para dar lugar a cualquier dispositivo imaginable, todos ellos con grados de miniaturización y conectividad cada vez mayores y a costes cada vez más bajos. Algunos ejemplos de dispositivos pueden ser Arduino, Spark, Intel Galileo o Waspmote, entre otros. Siendo Arduino, uno de los pilares del IoT actual, ya que permite que cualquier persona con conocimientos básicos de electrónica y programación pueda diseñar e implementar sus ideas de una forma sencilla.

Las redes: Dependiendo del proyecto de IoT que se quiera implementar, se dispondrá de

diferentes tipos de conexiones. Algunos proyectos pueden ser implementados dentro de una instalación que cuente con una conexión de red Ethernet-Wifi o de comunicación de objetos a corta distancia como RFID, Bluetooth, NFC. Mientras que otros proyectos, por razón de ubicación o movilidad, pueden requerir del uso de otras tipologías de red, ya sean privadas o públicas, como pueden ser GPRS, WiMAx, EDGE, 3G, 4G o combinadas.

Seguridad y privacidad: Este es un tema muy relevante en los proyectos IoT. Para realizar un proyecto IoT es importante pensar en aplicar los niveles necesarios de seguridad tanto a los dispositivos como a las plataformas. Algunos puntos que se deben tener en cuenta son:

- Que los dispositivos tengan capacidad de login seguro.
- Encriptación de las comunicaciones entre los dispositivos y las plataformas (gate-ways).
- Registros de dispositivos y autorizaciones especiales incluidos en los gateways.
- Inteligencia operacional para conocer si un usuario está haciendo uso de un servicio con un dispositivo a la vez en dos localizaciones distintas.

Los protocolos: El conocimiento de los distintos protocolos permitirá tomar mejores decisiones a la hora de adaptar las comunicaciones entre dispositivos y gateways. Entre los emergentes se tiene:

- MQTT/MQTTS: Orientado a dispositivos de baja fiabilidad y que consumen muy poco ancho de banda. Permite implementar un modelo bidireccional bajo publicación o subscripción.
- CoAP: Implementa un modelo equivalente a HTTP bajo UDP, eliminando la necesidad de sesión.
- REST API: Mayoritariamente utilizado para comunicaciones síncronas, tanto con con-centradores como con gateways.

3.1.3 REpresentational State Transfer (REST)

REST es un estilo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP, de ese modo se consigue crear APIs que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP (Asier Marqués, 2013). Hay que aclarar que REST no es una

tecnología, ni siquiera una arquitectura en sí, sino una familia de arquitecturas. Cualquier arquitectura de servicios distribuidos que cumpla con una serie de requisitos se puede considerar como una arquitectura REST (Enrique Amodeo, 2010).

A la hora de desarrollar una API REST, se debe tener en cuenta, el uso correcto de URIs, el uso correcto de HTTP e implementar Hypermedia. Además de esto, nunca se debe guardar estado en el servidor, toda la información que se requiere para mostrar el resultado debe estar en la consulta por parte del cliente. Esto significa, nada de variables de sesión en el servidor (JosmanTek, 2014). Para que una API REST nos recuerde, debemos identificarnos en cada llamada, ya sea a través de un usuario y una contraseña, un token u otro tipo de credencial. Lo mismo se debe hacer cada vez que se quiera acceder a cualquier tipo de información.

3.1.3.1 Uso correcto de URIs

Las URL (Uniform Resource Locator), son un tipo de URI (Uniform Resource Identifier) que permite identificar de forma única un recurso. Cada página, información en una sección, archivo, cuando se habla de REST, se nombran como recursos. El recurso por lo tanto es la información a la que se quiere acceder, modificar o borrar, independientemente de su formato. Cada URL de nuestra API es un recurso, ya sean datos o archivos.

La estructura básica de cada URL sería la siguiente:
{Protocolo}://{dominio o hostname}[:puerto (opcional)]/{ruta del recurso}?{consulta de filtrado}

Existen varias reglas básicas para ponerle nombre a la URI de un recurso (JosmanTek, 2014):

- Los nombres de URI no deben implicar una acción, por lo tanto debe evitarse usar verbos en ellos. Por ejemplo: /motos/2/editar sería incorrecto ya que se usa el verbo “editar”, la URI correcta seria /motos/2 independientemente de si se va a ver, editar o borrar.
- Deben ser únicas, no se debe tener más de una URI para identificar un mismo recurso.

- Deben ser independiente de formato. Por ejemplo: /motos/2.html sería incorrecto, se debe usar la misma URI ya sea html, txt, xml, o json.
- Deben mantener una jerarquía lógica. Por ejemplo: /motos/2/propietarios/1 no sería correcto, ya que es la moto la que pertenece al propietario, lo correcto sería /propietarios/1/motos/2
- Los filtrados de información de un recurso no se hacen en la URI. Para filtrar u ordenar se debe usar parámetros HTTP sobre la URI. Por ejemplo: /motos?orden=DESC&pagina=4

3.1.3.2 Uso correcto de HTTP

Conocer los métodos HTTP, códigos de estado y aceptación de tipos de contenidos es fundamental para desarrollar APIs REST.

Métodos HTTP

Con los métodos se puede realizar varias acciones sobre una misma URI (JosmanTek, 2014):

- **GET:** Para consultar y leer recursos. Ejemplo: GET /motos o GET /motos/2
- **POST:** Para crear recursos. Ejemplo: POST /motos
- **PUT:** Para editar recursos. Ejemplo: PUT/motos/2
- **DELETE:** Para eliminar recursos. Ejemplo: DELETE/motos/2
- **PATCH:** Permite modificar cierta información de un recurso. Ejemplo: PATCH/motos/2

Códigos de estado HTTP.

Cuando se realiza una operación, es vital saber si dicha operación se ha realizado con éxito o en caso contrario, por qué ha fallado. Por ejemplo si una consulta tiene datos insuficientes no se puede devolver el resultado con estado 200, que significa que la operación se ha realizado correctamente. Es muy importante conocer estos códigos y saber cuándo utilizarlos, así lo correcto sería devolver el error anterior con el estado HTTP 400, solicitud incorrecta. A continuación se describirá de forma general cada uno de los códigos de estado HTTP.

El primer dígito del código de respuesta específica una de las cinco clases de respuesta

(Wikipedia, 2015, Códigos de estado HTTP):

- **1xx: Respuestas informativas.** Esta respuesta significa que el servidor ha recibido los encabezados de la petición, y que el cliente debería proceder a enviar el cuerpo de la misma
- **2xx: Peticiones correctas.** Esta clase de código de estado indica que la petición fue recibida correctamente, entendida y aceptada.
- **3xx: Redirecciones.** El cliente tiene que tomar una acción adicional para completar la petición.
- **4xx Errores del cliente.** La solicitud contiene sintaxis incorrecta o no puede procesarse.
- **5xx Errores de servidor.** El servidor falló al completar una solicitud aparentemente válida.

Tipos y formatos de contenido.

Cuando se habló de URLs, se dijo que no era correcto indicar el tipo de formato de un recurso al cual se desee acceder o manipular.

HTTP permite especificar en qué formato se quiere recibir el recurso, pudiendo indicar varios en orden de preferencia, para ello se utiliza el headerAccept. La API devolverá el recurso en el primer formato disponible y, de no poder mostrar el recurso en ninguno de los formatos indicados por el cliente mediante el header Accept, devolverá el código de estado HTTP 406. En la respuesta, se devolverá el header Content-Type, para que el cliente sepa qué formato se devuelve, por ejemplo: “application/pdf” para devolver un PDF (Asier Marqués, 2013).

3.1.3.3 Implementar Hypermedia

Implementar Hypermedia consiste en conectar mediante vínculos las aplicaciones clientes con las APIs, permitiendo a dichos clientes despreocuparse por conocer de antemano cómo acceder a los recursos. Con Hypermedia básicamente se añade información extra al recurso sobre su conexión a otros recursos relacionados con él (Asier Marqués, 2013).

3.1.4 Protocolo MQTT

MQTT (Message Queue Telemetry Transport), es un protocolo usado para la

comunicación **machine-to-machine** (M2M) en el **"Internet of Things"**, ya que se ha diseñado para ser un protocolo de mensajería extremadamente ligero basado en TCP/IP. Este protocolo está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos embebidos con pocos recursos como Arduino, además de ser también ideal para aplicaciones móviles por su envío eficiente (Luis Miguel Gracia, 2013).

Se usa por ejemplo en el Facebook Messenger para iPhone y Android.

La arquitectura de MQTT sigue una topología de estrella, con un nodo central que hace de servidor o "broker" con una capacidad de hasta 10000 clientes. El broker es el encargado de gestionar la red y de transmitir los mensajes, para mantener activo el canal, los clientes mandan periódicamente un paquete (PINGREQ) y esperan la respuesta del broker (PINGRESP). La comunicación se basa en unos "**topics**" (temas), que el cliente que publica el mensaje crea y los nodos que deseen recibirlo deben subscribirse a él. La comunicación puede ser de uno a uno, o de uno a muchos (José Antonio Yébenes Gálvez, 2015).

Un "topic" se representa mediante una cadena y tiene una estructura jerárquica. Cada jerarquía se separa con '/'. Por ejemplo, "**mensajes/to/userName/motos/motoID**". De esta manera se pueden crear jerarquías de clientes que publican y reciben datos. Por lo tanto un nodo puede subscribirse a un "topic" concreto ("**mensajes/to/userName/motos/motoID**") o a varios ("**mensajes/to/userName/motos/#**"). Una característica muy importante del MQTT es que al ser un protocolo tan ligero existen clientes y servidores MQTT en diversos lenguajes, entre los cuales tenemos (Luis Miguel Gracia, 2012):

Como **Servidores**:

- **IBM WebSphere MQ Telemetry:** es un addon para MQ versión 7 y superiores.
- **Mosquitto:** Servidor opensource MQTT con clientes C,C++, Python y Javascript
- **Mosca.js:** un Servidor MQTT para node.js
- **Apache Apollo:** el nuevo ActiveMQ soporta MQTT via a plugin.

Como **APIS Clientes**:

- **Arduino client for MQTT**
- **Mosca.js:** un cliente MQTT para node.js

- **Eclipse Paho:** cliente Java
- **mosquitto Javascript / Websocket client**

La elección del servidor como del cliente MQTT dependerá de las características y necesidades de nuestro proyecto.

Por otro lado, la versión 3 del protocolo utiliza la opción de publicar/suscribir y da soporte a tres calidades de servicio que permiten configurar el nivel de seguridad de un mensaje de Publish: "como máximo una vez", "al menos una vez" y "exactamente una vez", las cuales se detallaran a continuación (IBM, 2015):

Como máximo una vez:

- QoS = 0 es el modo más rápido de la transferencia. A veces se llama "dispara y olvida".
- El mensaje se entrega como máximo una vez, o no se entrega en absoluto. Su entrega a través de la red no se reconoce.
- El mensaje no se almacena. El mensaje podría perderse si el cliente se desconecta, o si falla el servidor.
- El MQTT protocolo no requiere servidores para reenviar publicaciones en QoS = 0 a un cliente. Si el cliente se desconecta en el momento en que el servidor recibe la publicación, la publicación podría ser descartada, dependiendo del servidor.

Al menos una vez

- QoS = 1 es el modo por defecto de la transferencia.
- El mensaje se entrega siempre al menos una vez. Si el remitente no recibe un acuse de recibo, el mensaje se envía de nuevo con el DUP flag conjunto hasta que se reciba un acuse de recibo. Como resultado el receptor puede enviar y procesar el mismo mensaje varias veces.
- El mensaje debe ser almacenado localmente en el emisor y el receptor hasta que se procese.
- El mensaje se elimina del receptor después de que se ha procesado. Si el receptor es un broker, el mensaje se publica a sus suscriptores. Si el receptor es un cliente, el mensaje se entrega a la aplicación que lo suscribe. Después se elimina el mensaje, el receptor envía un acuse de recibo al remitente.

- El mensaje se elimina del remitente después de haber recibido un acuse de recibo del receptor.

Exactamente una vez

- QoS = 2 es el modo más seguro, pero más lento de transferencia. Se necesitan al menos dos pares de las transmisiones entre el emisor y el receptor antes de eliminar el mensaje del remitente. El mensaje puede ser procesado en el receptor después de la primera transmisión.
- El mensaje siempre se entrega exactamente una vez.
- El mensaje debe ser almacenado localmente en el emisor y el receptor hasta que se procese.
- En el primer par de transmisiones, el emisor transmite el mensaje y obtiene el acuse de recibo desde el receptor que ha almacenado el mensaje. Si el remitente no recibe un acuse de recibo, el mensaje se envía de nuevo con el DUP flag conjunto hasta que se reciba un acuse de recibo.
- En el segundo par de transmisiones, el remitente le indica al receptor que se puede completar el procesamiento del mensaje, "PUBREL". Si el remitente no recibe un acuse de recibo del mensaje "PUBREL", este último se envía de nuevo hasta que se reciba un acuse de recibo. El remitente borra el mensaje guardado cuando se recibe el acuse de recibo del mensaje "PUBREL".
- El receptor puede procesar el mensaje en la primera o segunda fases, siempre que no vuelva a procesar el mensaje. Si el receptor es un broker, se publica el mensaje a los suscriptores. Si el receptor es un cliente, se entrega el mensaje a la aplicación que lo suscribe. El receptor envía un mensaje de finalización al remitente avisando que se ha terminado de procesar el mensaje.

Nuestro proyecto está relacionado con el Internet de las cosas, debido a esto, se ha optado por utilizar este protocolo porque es uno de los más implementados en este tipo de proyectos y además porque cubre de manera eficiente todos los requerimientos de nuestro proyecto en cuanto a:

- Comunicación machine-to-machine (Móvil del usuario y dispositivo electrónico).

- Al ser un protocolo muy ligero puede ser utilizado en dispositivos que cuentan con pocos recursos (CPU, RAM, etc.), en nuestro caso Arduino y sus módulos.
- Consume muy poco ancho de banda indispensable para la comunicación via GPRS del dispositivo.
- Además de ser también ideal para nuestra aplicación móvil por su envío eficiente de mensajes.

A continuación se puede observar como se utiliza el protocolo MQTT en el sistema de seguridad para motocicletas (**SisMo**):

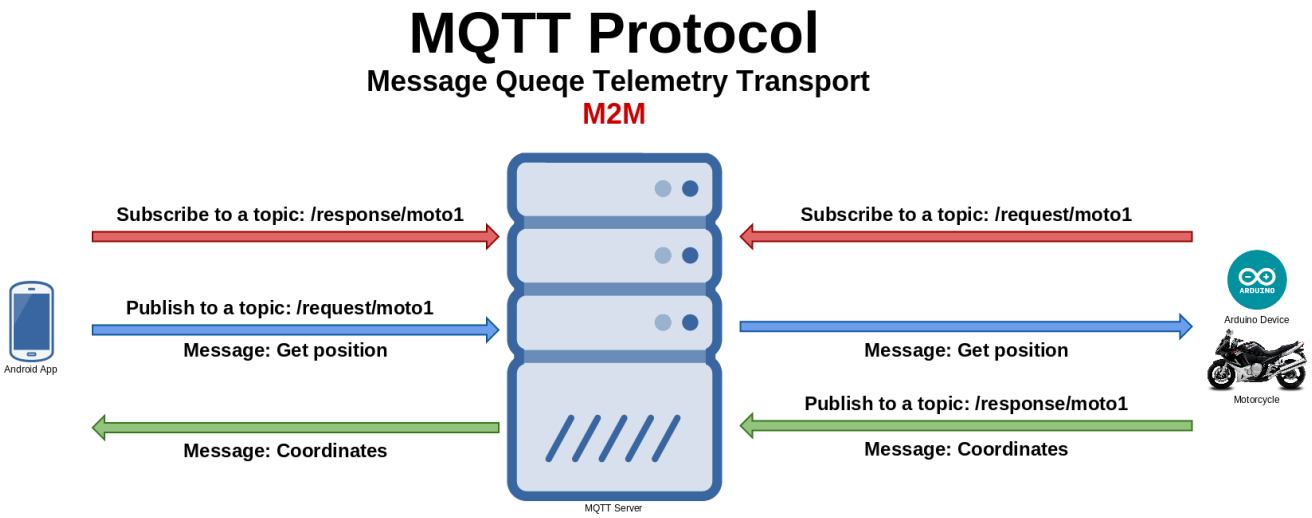

Figura 5: Envió y recepción de mensajes 1 a 1

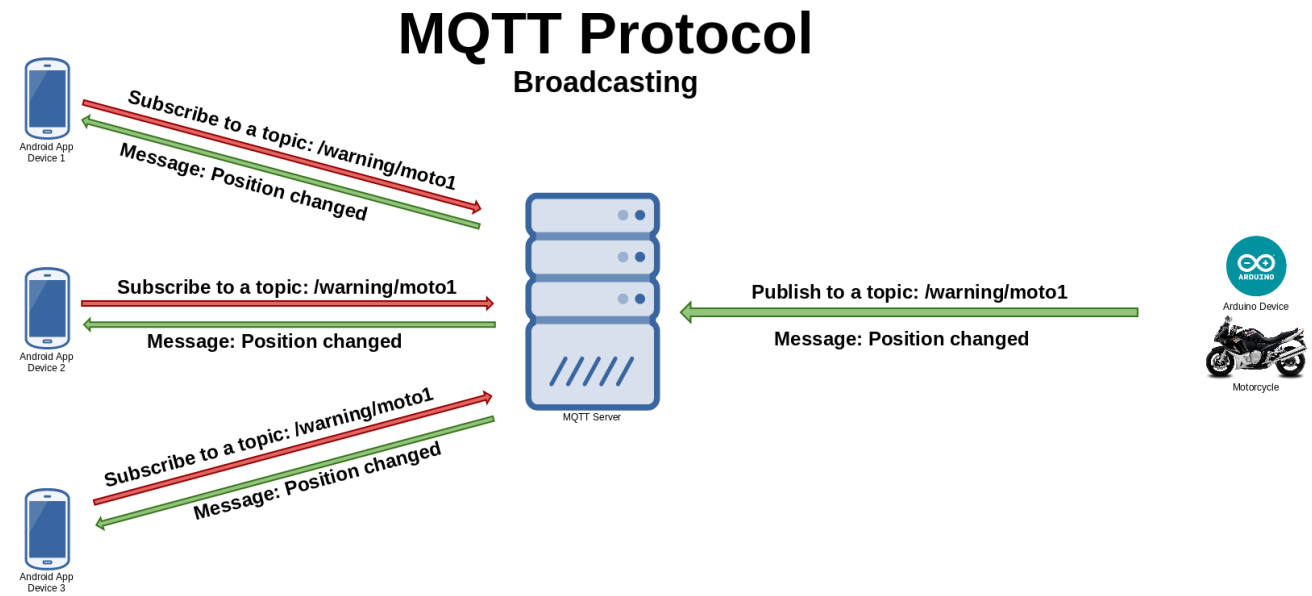

Figura 6: Envió y recepción de mensajes 1 a muchos

3.1.5 Arduino

Arduino es una plataforma de prototipos electrónica de código abierto, basada en hardware y software flexibles (Rafael Enríquez Herrador, 2009). Arduino cuenta con una gran variedad de modelos, cada uno pensado para un público concreto o para una serie de tareas específicas. En este documento se va a detallar específicamente el modelo correspondiente a Arduino Leonardo que es componente principal del dispositivo electrónico de SisMo y que se puede observar en la siguiente imagen:

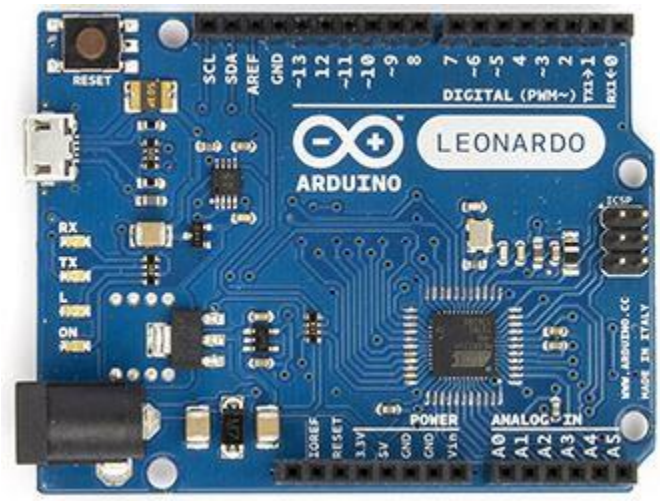

Figura 7: Arduino Leonardo

El **Arduino Leonardo** es una placa electrónica basada en el ATmega32u4. Cuenta con 20 pines digitales de entrada / salida (de los cuales 7 se pueden utilizar como salidas PWM y 12 como entradas analógicas), un cristal oscilador a 16 MHz, una conexión micro USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para utilizar el microcontrolador; para empezar a trabajar con él, simplemente se conecta a un ordenador a través del cable USB o alimentándolo con un adaptador AC/DC o una batería. El Arduino Leonardo se diferencia de todas las otras placas anteriores, por el hecho de que el ATmega32u4 tiene ya en su interior la parte de comunicación USB, así que se elimina la necesidad de un microprocesador secundario. Esto permite que Leonardo, una vez conectado a un ordenador, se reconozca como un ratón o un teclado, y también emula permanentemente un puerto de serie (CDC) / COM (Arduino, 2015).

Descripción técnica

Microprocesador	ATmega32u4
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Digital pines I / O	20
Canales PWM	7
Canales de entrada analógicos	12
Corriente Máxima para Pines E/S	40 mA
Corriente Máxima para Pines 3.3V	50 mA
Memoria flash	32 KB (ATmega32u4) de los cuales 4 KB son utilizados para el gestor de arranque
SRAM	2,5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad de reloj	16 MHz
Largo	68,6 mm
Ancho	53,3 mm
Peso	20g

Figura 8: Datos técnicos del Arduino Leonardo

Programación

El Leonardo puede ser programado con el software Arduino. El ATmega32U4 del Arduino Leonardo viene con un bootloader pregrabado que permite cargar nuevo código a él sin el uso de un programador de hardware externo. Se comunica usando el protocolo AVR109. También se puede saltar el bootloader y programar el microprocesador a través de la cabecera ICSP (In-Circuit Serial Programming).

3.1.6 Sistema de posicionamiento global (GPS)

GPS de (Global Positioning System) o Sistema de Posicionamiento Global en español, es un sistema que permite determinar en todo el mundo la posición e incluso la velocidad de un objeto o una persona. El GPS está compuesto por una red de 24 satélites en órbita sobre el planeta tierra, a 20.200 km de altura, con trayectorias sincronizadas con el fin de cubrir toda la superficie de la tierra y encargados de triangular la posición de cualquier objeto que le pida información a estos (Wikipedia, 2014, Sistema de Posicionamiento Global).

Para determinar la posición de un receptor, el sistema GPS hace uso de la trilateración, que funciona de la siguiente manera:

- Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio la distancia total hasta el receptor.
- Obteniendo información de dos satélites queda determinada una circunferencia que resulta cuando se intersecan las dos esferas en algún punto de la cual se encuentra el receptor.
- Teniendo información de un tercer satélite, se elimina el inconveniente de la falta de sincronización entre los relojes de los receptores GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS puede determinar una posición 3D exacta: **latitud, longitud y altitud**.

Hay que tener en cuenta que los receptores GPS manejan un rango de error que va desde centímetros hasta metros, pues su precisión dependerá de la posición del satélite y del retraso de la señal.

El GPS es una de las tecnologías fundamentales en el sistema desarrollado (SisMo), ya que le permitirá a los usuarios conocer: la posición de su motocicleta en cualquier momento, si esta ha sido desplazada y en caso de robo poder rastrearla.

3.1.7 Sistemas de Tiempo Real

Un sistema de tiempo real (STR) es un sistema informático que interacciona repetidamente con su entorno físico y responde a los estímulos que recibe del mismo dentro de un plazo de tiempo determinado (Juan Antonio de la Puente, 2007). Por lo anterior, su eficiencia no solo depende de la exactitud de los resultados de cómputo, sino también del intervalo de tiempo especificado en que tienen que ejecutarse. El tiempo en que se ejecutan las acciones del sistema es significativo.

A diferencia de los sistemas tradicionales, que tienden a distribuir en forma equitativa los recursos disponibles entre las diferentes tareas a ejecutar, los sistemas de tiempo real deben asegurar la distribución de recursos de tal forma que se cumplan los requerimientos de tiempo (Alejandro L. Veiga, S.F).

Un ejemplo común de un STR es la red social Facebook, la cual informa de manera prácticamente inmediata, cualquier cambio que haya realizado en su estado una persona con la que tenemos contacto. Es decir, cuando un usuario actualiza su estado en Facebook, inmediatamente la plataforma es capaz de mostrarle a todos sus amigos el cambio que ha realizado; esto se realiza en fracciones de segundos, lo cual da una sensación de tiempo real.

SisMo está diseñado y desarrollado para ser un sistema de tiempo real, ya que la funcionalidad del mismo lo amerita. Este permitirá la intercomunicación y él envió de respuestas de forma inmediata entre cada uno de los componentes del sistema y entre estos y el usuario. Al decir “de forma inmediata” nos referimos a fracciones de segundos, porque como tal la inmediatez aun no es posible.

3.2 MARCO CONCEPTUAL

3.2.1 Botón o Pulsador

Un botón o pulsador es un dispositivo utilizado para realizar cierta función. Los botones son de diversas formas y tamaño y se encuentran en todo tipo de dispositivos, aunque principalmente en aparatos eléctricos y electrónicos.

Los botones son por lo general activados, al ser pulsados con un dedo. Permiten el flujo de corriente mientras son accionados. Cuando ya no se presiona sobre él vuelve a su posición de reposo.
Puede ser un contacto normalmente abierto en reposo NA o NO (Normally Open en Inglés), o con un contacto normalmente cerrado en reposo NC (Wikipedia, 2013).

3.2.2 Relé

El relé o relevador es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes (Wikipedia, 2013).

3.2.3 Baquelita

La baquelita fue la primera sustancia plástica totalmente sintética, creada en 1907 por Leo Baekeland. Se trata de un fenoplástico que actualmente tiene aplicaciones de interés. Lo sintetizó a partir de moléculas de fenol y formaldehído. Este producto puede moldearse a medida que se forma y resulta duro al solidificar. No conduce la electricidad, es resistente al agua y los disolventes, pero fácilmente mecanizable (Tecnología de los Plásticos ,2011). Es el material más utilizado para la fabricación de circuitos impresos.

3.2.4 Motocicleta/Moto

Una motocicleta es un medio de dos ruedas impulsado por un motor de combustión interna a gasolina. El cuadro y las ruedas constituyen la estructura fundamental del vehículo. La rueda directriz es la delantera y la rueda motriz es la trasera; actualmente la motocicleta es considerada como un medio de transporte alternativo y por su bajo costo a comparación de otros medios de trasporte (carro), está al alcance de cualquier ciudadano (Oscar Vega Pava, 2012).

3.2.5 Dispositivo móvil

Es un dispositivo construido sobre una plataforma informática móvil capaz de realizar actividades semejantes a una computadora pero sin los problemas de deslazamiento que estas conllevan y con una mayor conectividad.

Entre los rasgos comunes de un dispositivo móvil está la función multitarea, el acceso a internet vía Wi-Fi o red 3G, función multimedia (cámara y reproductor de vídeos/mp3), a los programas de agenda, administración de contactos, acelerómetros, GPS y algunos programas de navegación, así como ocasionalmente la habilidad de leer documentos de negocios en variedad de formatos como PDF (Wikipedia, 2014, Dispositivo móvil).

3.2.6 Aplicación web

Una aplicación web es una interfaz a la que se accede a través de un navegador web como Chrome, Firefox o Internet Explorer, la cual puede prestar distintas funcionalidades, dependiendo en el ámbito en el que se muestra (EcuRed, 2014). Por ejemplo una aplicación web puede ser informativa, si es un aplicación hecha por una empresa de comunicaciones, como un canal de televisión, puede ser social cuando se encarga de conectar persona como es el caso de Facebook, puede ser para entretenimiento cuando a través de dicha aplicación se pueden jugar o ver vídeos como es el caso de Youtube.

Las aplicaciones web son populares debido a que funcionan a través de navegadores y esto les da independencia del sistema operativo, así como la facilidad para actualizarse y mantenerse sin necesidad de instalar software adicional.

3.2.7 Aplicación móvil

Una aplicación móvil o app móvil es una interfaz nativa que se le presenta a un usuario en un dispositivo móvil, estas aplicación son desarrolladas mayormente cuando un sistema necesita tener acceso total a los recursos de un dispositivo móvil (recursos a los que un navegador web no podría acceder), y por tanto no sería factible desarrollar una aplicación web, es por esto que muchas empresas que iniciaron con aplicaciones web, han optado por tener también una aplicación móvil que se desempeñe en un campo especifico en la cual la aplicación web no sería la mejor opción. Por lo general estas aplicaciones se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, Windows Phone, entre otros (Wikipedia, 2014, Aplicación móvil).

3.2.8 Android O.S

Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para

dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas. Este sistema operativo será usado como plataforma de desarrollo para las aplicaciones móviles. Se optó por Android como plataforma principal de desarrollo porque este es el sistema operativo con mayor demanda en el mercado y tiene una buena y basta documentación la cual podría ser indispensable al momento de desarrollar las aplicaciones (Alejandro Nieto Gonzales, 2011).

3.2.9 Navegador web

Un navegador web, o browser, es un software que permite el acceso a Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos (Wikipedia, 2014, Navegador web). En la actualidad, hay un gran variedad de navegadores que funcionan en diferentes plataformas tanto móviles como de escritorio, entre estos navegados tenemos: Google Chrome, Firefox, Internet Explorer, Opera, Safari, etc.

3.2.10 Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar estas peticiones de forma adecuada, entregando como resultado una página web o cualquier otro tipo de información dependiendo de lo que solicite el cliente (Mis respuestas, 2012).

3.2.11 Bases de datos

Una base de datos es un “almacén” que permite guardar grandes cantidades de información de forma organizada para luego poder acceder a ella fácilmente. Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos (Damián Pérez Valdés, Maestros del web, 2007).

3.2.12 Node.js

Es una plataforma para construir aplicaciones en JavaScript la cual corre sobe el motor V8 de Google Chrome, en la actualidad es muy utilizado para construir aplicaciones altamente escalables. Node.js utiliza un modelo no bloqueante de entrada/salida dirigida por eventos lo cual lo hace ligero y eficiente, ideal para aplicaciones en tiempo real, y aplicaciones con uso

intensivo de datos que se ejecutan a través de dispositivos distribuidos (NodeJS, 2014).

3.2.13 MongoDB

Es una base de datos no relacional de código abierto orientado a documento para su fácil desarrollo y escalabilidad. MongoDB ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidatos. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria (in-memory). La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa (MongoDB, 2014).

3.2.14 Java

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

3.2.15 JavaScript

Abreviado comúnmente "JS", es un lenguaje de programación interpretado. Se define como: orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente (client-side), implementado como parte de un navegador web, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

4. METODOLOGIA

El tipo de investigación seleccionado para el desarrollo de este proyecto, corresponde a una investigación aplicada de naturaleza descriptiva, debido a que en primer lugar se aplicaron los conocimientos adquiridos durante la investigación a la práctica (desarrollo de un sistema de seguridad para motocicletas), contribuyendo a resolver un problema social (hurto de motocicletas en Colombia). En segundo lugar se describen aspectos relevantes del problema investigado, buscando exponer la realidad en que este se desenvuelve.

4.1 Fases del proyecto

4.1.1 Fase de investigación documental

Esta fase se centró en los procesos de búsqueda, recolección, selección y análisis de información relevante para el tema de investigación, utilizando el internet como principal herramienta de búsqueda y accediendo a fuentes directas de información como libros, artículos, informes y sitios web pertenecientes a organizaciones estatales, empresas y autores independientes.

Con la realización de esta fase la intención fue adquirir conocimientos y fundamentos necesarios para lograr un enfoque adecuado del tema de investigación así como para hacer un bosquejo inicial de los requerimientos de hardware y software necesarios para el diseño y desarrollo del sistema. Para esto, se investigó el funcionamiento y manejo de la placa Arduino y del módulo GSM/GPRS/GPS, tanto en software como en hardware. Se averiguó la manera cómo funciona la parte mecánica y eléctrica de la motocicleta donde se implementa el dispositivo, de este mismo modo se indagó sobre la infraestructura necesaria para la comunicación de los diferentes componentes del sistema, lo que conllevó a adquirir conocimientos en programación para Android, funcionamiento del protocolo MQTT, configuración de un servidor web y manejo de base de datos.

4.1.2 Fase de análisis de requerimientos

En esta fase se realizó un análisis general de cómo debería funcionar el sistema. Este análisis arrojo los siguientes requerimientos el cual se dividen en funcionales y no funcionales.

4.1.2.1 Requerimientos funcionales

- El usuario podrá registrase en el sistema.
- El usuario podrá iniciar sesión en el sistema.
- El usuario podrá registrar motocicletas a su cuenta en el sistema.
- El usuario podrá ver sus motocicletas registradas en su cuenta.
- El usuario podrá activar y desactivar desde su aplicación móvil el monitoreo de cualquiera de sus motocicletas registradas.
- El usuario podrá obtener desde su aplicación móvil el estado de los elementos monitoreados de cualquiera de sus motocicletas registradas.
- El usuario podrá obtener desde su aplicación móvil la posición de cualquiera de sus motocicletas registradas.
- El dispositivo electrónico debe permitir el bloqueo y desbloqueo de sistema eléctrico de la motocicleta.
- El dispositivo electrónico debe verificar el estado del seguro de la motocicleta.
- El dispositivo electrónico debe permitir saber la ubicación de la motocicleta.
- Si el monitoreo está activo, el dispositivo electrónico debe detectar y notificar al usuario cualquier alteración en el seguro de la motocicleta.
- Si el monitoreo está activo, el dispositivo electrónico debe detectar cuando la motocicleta es desplazada.

4.1.2.2 Requerimientos no funcionales

- La aplicación del sistema estará diseñada para funcionar en sistemas operativos Android.
- El sistema necesitara conexión permanente a internet.
- El sistema debe estar en capacidad de permitir en el futuro la escalabilidad es decir la modificación, eliminación y el desarrollo de nuevas funcionalidades.

- Usabilidad: La aplicación se debe diseñar de manera que la interfaz sea lo más amigable posible.
- El sistema deberá rechazar accesos o modificaciones no autorizadas a este.
- El tiempo de respuesta de la aplicación tendrá que ser eficiente.
- El consumo de datos por parte del dispositivo electrónico y la aplicación móvil tendrá que ser óptimo.

4.1.3 Fase de diseño del sistema

En esta fase se describe de forma general el sistema y se toman las consideraciones pertinentes en cuanto al hardware y software necesarios para el desarrollo de este, así como también se representan las funciones del sistema a través de bosquejos y de diagramas como: casos de usos, diagramas de secuencia, diagramas de clases y diagramas de componentes.

Se realiza el diseño de la interfaz de usuario, el diseño de la base de datos y los circuitos electrónicos necesarios, teniendo en cuenta que estos se adapten de la mejor manera a las necesidades funcionales del sistema.

Todo lo anterior está plasmado detalladamente en el ítem **5.1 Diseño del sistema** de este documento.

4.1.4 Fase de desarrollo

En esta fase se transformaron todos los diseños realizados previamente a código y se construyen físicamente los circuitos electrónicos necesarios para el funcionamiento del dispositivo, es decir, se inicia el proceso de construcción y programación de cada uno de los módulos que integran el sistema, tanto web, móvil y electrónicos haciendo uso de lenguajes de programación como: Java, JavaScript y processing, buscando darle contenido y funcionalidad al sistema.

También se crearon las bases de datos necesarias para la plataforma, se configuraron los

distintos servidores, se implementó el dispositivo en la motocicleta y se realizaron pruebas unitarias de cada uno de los módulos o componentes pertenecientes al sistema.

El contenido de esta fase se puede observar detalladamente en el punto **5.2 Desarrollo** de este documento.

4.1.5 Fase de pruebas

Una vez terminada la fase de desarrollo y al integrarse los distintos módulos que componen el sistema, se realizaron pruebas para evaluar la interacción y respuesta de este, con el fin de asegurar la eficiencia y el correcto funcionamiento del sistema. Para esto realizamos pruebas con el fin de garantizar que el sistema funcione correctamente y sin ningún inconveniente.

Las pruebas realizadas y los resultados obtenidos en esta fase se pueden observar en el punto **6. Pruebas y resultados** de este documento.

4.2 Metodología de desarrollo del producto

Para el desarrollo del sistema propuesto (**SisMo**) y para cumplir con cada uno de los objetivos propuestos en este proyecto en los tiempos requeridos, se ha optado con implementar una metodología de desarrollo ágil conocida como **SCRUM**.

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación (SOFTENG, 2015).

Se optó por tomar SCRUM como metodología para desarrollar el sistema propuesto ya nos brinda los siguientes beneficios (SOFTENG, 2015):

- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado.
- **Mayor calidad del software:** La metodología de trabajo y la necesidad de obtener

una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.

- **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- **Predicciones de tiempos:** Mediante esta metodología se conoce la velocidad media del equipo, con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad.
- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

5. DISEÑO Y DESARROLLO

5.1 Diseño del sistema

5.1.1 Arquitectura del sistema

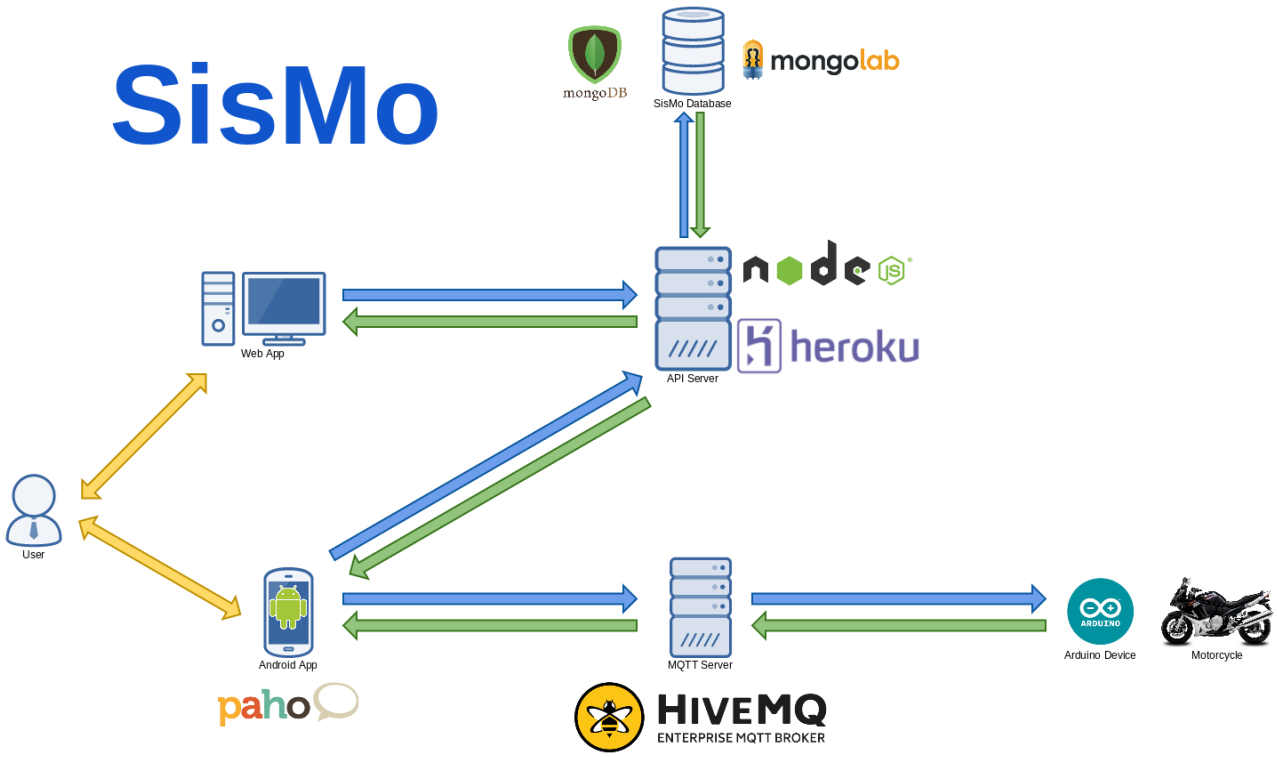

Figura 9: Arquitectura del sistema (SISMO)

En la fig.9 se puede apreciar la arquitectura general del sistema (**SisMo**), la cual consta de:

- Una aplicación móvil desarrollada de forma nativa para el sistema operativo **Android**.
- Una aplicación web desarrollada con tecnologías **HTML5**, **CSS3** y **Javascript**.
- Un dispositivo electrónico construido con base a la plataforma **Arduino**.
- Un servidor web desarrollado con **Node.js** y alojado en un hosting de **Heroku**.
- Un servidor MQTT alojado en un hosting de **HIVEMQ**.
- Una base de datos **MongoDB** alojada en un hosting como **MongoLab**

La interacción de todos estos componentes será detallada en el ítem **5.1.2 Diagrama de componentes**

5.1.2 Diagrama de componentes

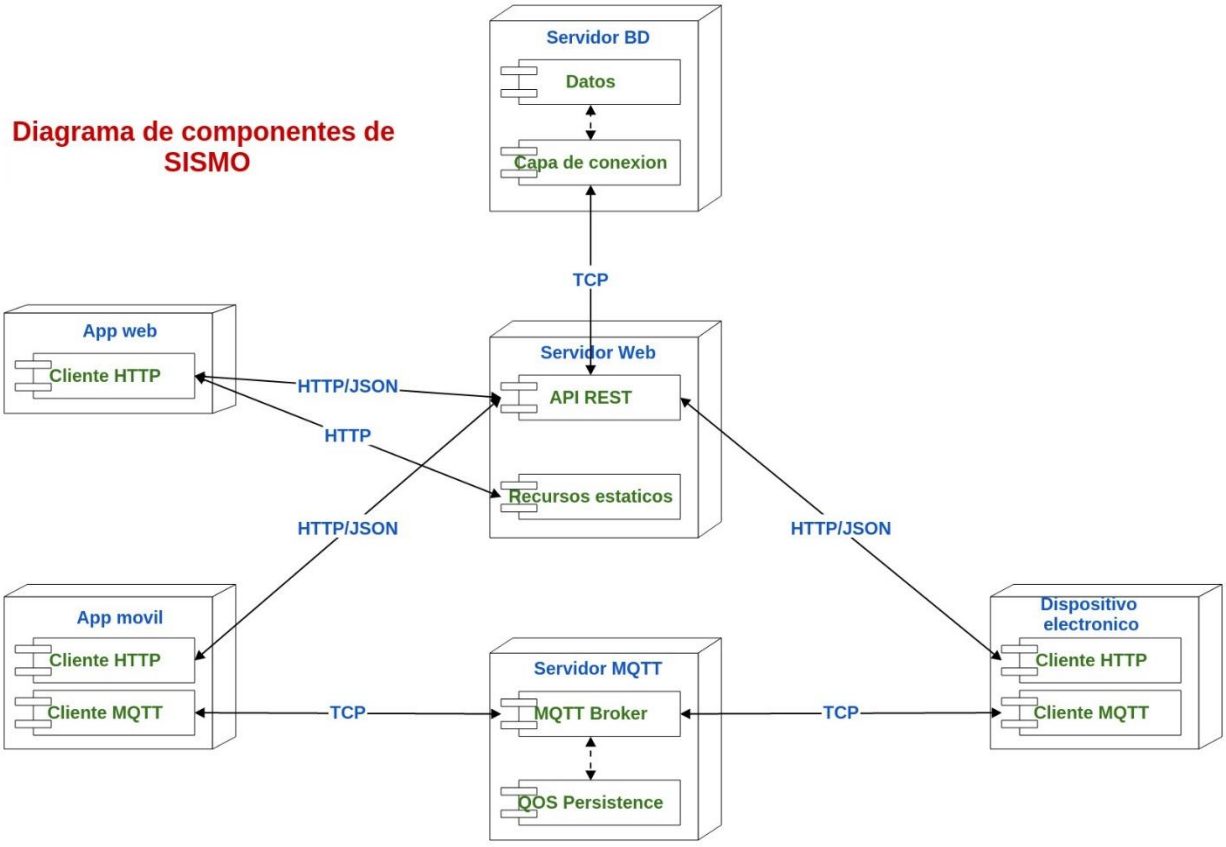

Figura 10: Diagrama de componentes

En el diagrama anterior se puede observar lo siguiente:

- La **Aplicación Web** realiza a través del protocolo **HTTP** peticiones al **Servidor Web** el cual (dependiendo si está pidiendo un recurso a la **API** o un **Recurso estático**), le enviara respuestas a través del mismo protocolo.

- La **Aplicación Móvil** realizara dos tipos de conexiones:

    - La primera conexión la realiza a través del protocolo **HTTP** hacia la **API** en el **Servidor Web**, el cual le enviara respuesta utilizando el mismo protocolo, esta conexión la realiza para obtener todos los datos de las motocicletas que el usuario tiene registrada.

    - La segunda conexión la realiza utilizando el protocolo **MQTT** (el cual utiliza TCP como protocolo de conexión a bajo nivel) cuando se conecta al **MQTT Broker**

en el **Servidor MQTT**, esto lo realiza para establecer conexión con las motocicletas registradas en el sistema, enviar acciones a las motocicletas y recibir notificaciones de las mismas

- La **Dispositivo Electrónico** realizara dos tipos de conexiones:
    - La primera conexión la realiza a través del protocolo **HTTP** hacia la **API** en el **Servidor Web**, esta conexión la hace para actualizar el estado de la motocicleta en cualquier momento.
    - La segunda conexión la realiza utilizando el protocolo **MQTT** (el cual utiliza TCP como protocolo de conexión a bajo nivel) cuando se conecta al **MQTT Broker** en el **Servidor MQTT**, esto lo realiza para establecer conexión con las aplicaciones móviles que tiene el dueño de la motocicleta, recibir acciones de las aplicaciones y enviar notificaciones a las mismas.
- El **Servidor MQTT** implementa una capa de calidad de servicios **QOS**, esto con el fin de brinda una funcionalidad completa del protocolo **MQTT.**
- Si es necesario el **Servidor Web** se conectara a un **Servidor de Bases de Datos** para obtener los datos que se estén pidiendo a la **API.** Esta conexión se realiza a través del protocolo **TCP**.

5.1.3 Diagramas de casos de usos

5.1.3.1 Diagrama general

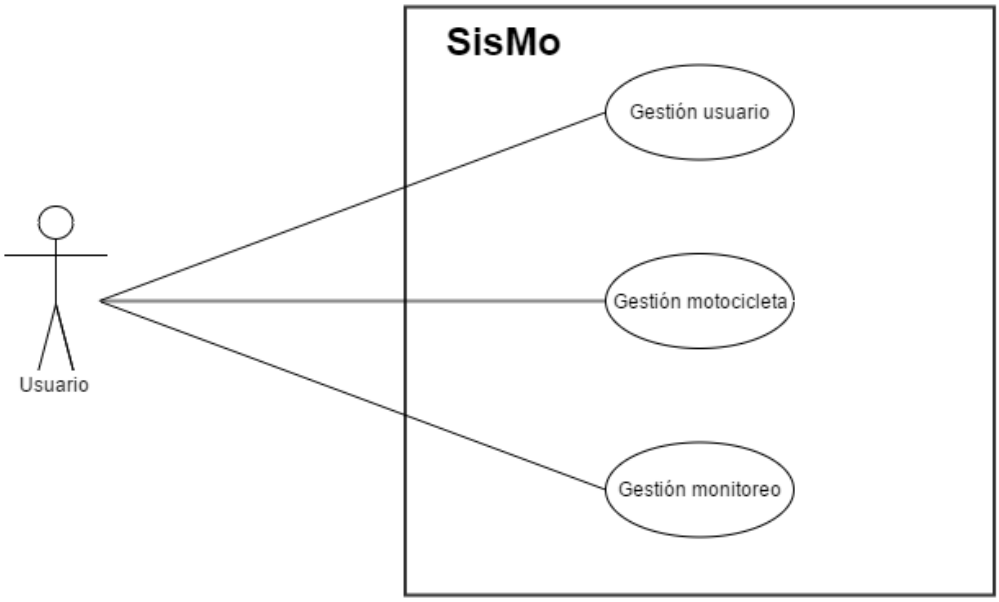

Figura 11: C.U - Diagrama general

En la Fig. 11 se muestra de manera general los tres tipos de gestión que puede desarrollar un usuario en el sistema de seguridad para motocicletas (**SisMo**). Cada una de estas se detallaran a través de su respectivo caso de uso.

5.1.3.2 Gestión usuario

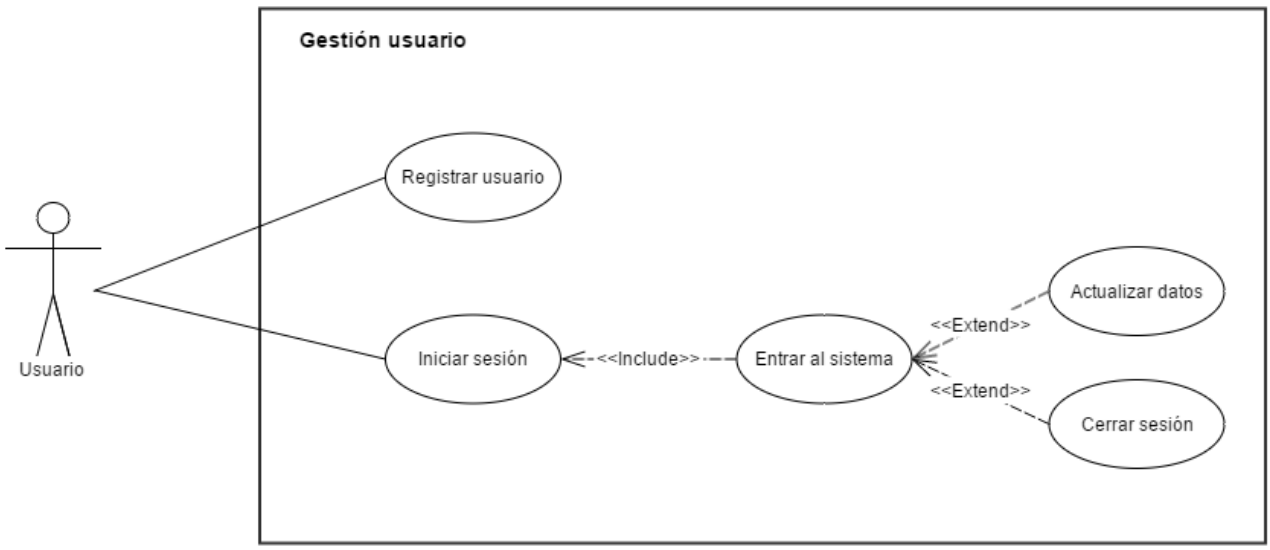

Figura 12: C.U – Gestión usuario

En la Fig. 12 se representa la “Gestión usuario” llevada a cabo por una persona en el sistema y destinada exclusivamente a las funciones más básicas desempeñadas por este. Un usuario no registrado debe ingresar sus datos al sistema los cuales son validados por el mismo, si los datos son válidos, el usuario pasa a ser un usuario registrado obteniendo el permiso para entrar al sistema. Una vez que el usuario ha iniciado sesión, podrá: actualizar sus datos y cerrar su sesión.

5.1.3.3 Gestión motocicleta

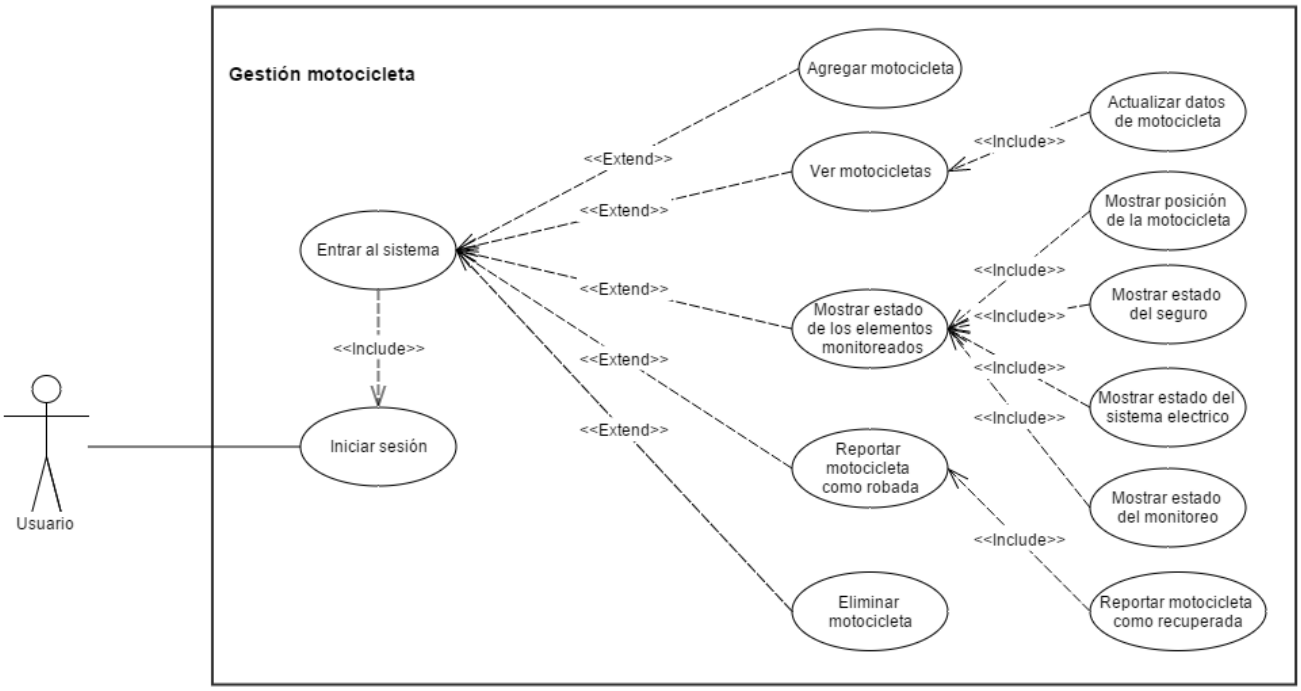

Figura 13: C.U - Gestión motocicleta

En la Fig. 13 se representa la “Gestión motocicleta” llevada a cabo por una persona en el sistema y destinada exclusivamente a las funciones relacionadas con las motocicletas del usuario. Una vez que el usuario ha iniciado sesión, podrá: Agregar motocicletas, ver sus motocicletas registradas y actualizar los datos de las mismas, mostrar el estado de los elementos monitoreados como: posición de la motocicleta, seguro, sistema eléctrico y el estado del monitoreo; reportar una motocicleta especifica como robada o recuperada, y por ultimo eliminar las motocicletas.

5.1.3.4 Gestión monitoreo

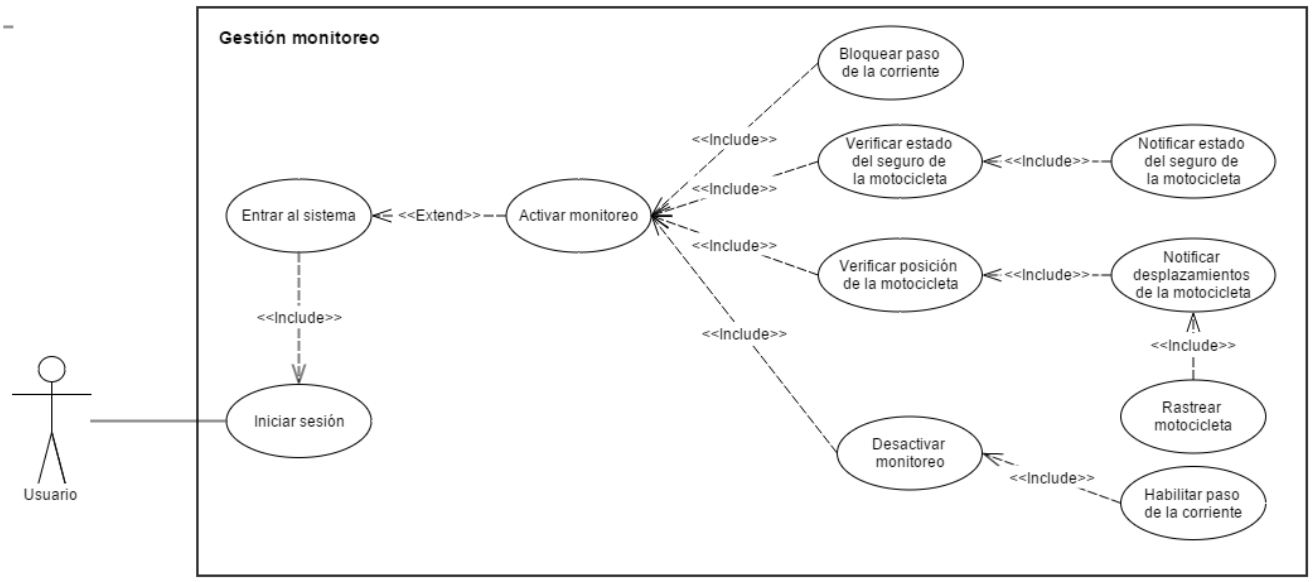

Figura 14: C.U - Gestión monitoreo

En la Fig. 14 se representa la “Gestión monitoreo” llevada a cabo por una persona en el sistema y destinada exclusivamente a las funciones que se desencadenan al activar el monitoreo del sistema. Una vez que el usuario ha iniciado sesión, podrá activar el monitoreo. Cuando el usuario activa el monitoreo, automáticamente el dispositivo electrónico bloquea el paso de la corriente (No se puede encender la motocicleta) y se encarga de verificar constantemente el estado del seguro y la posición de la moto. Si el dispositivo electrónico nota algún cambio ya sea en el seguro o en la posición de la moto, inmediatamente la aplicación móvil se lo notificará al usuario. En caso de que la motocicleta se desplace más de 25 metros de su posición inicial, la aplicación mostrará la opción para el rastreo en tiempo real de la motocicleta; por último, el usuario también podrá desactivar el monitoreo del sistema y con esto habilitar el paso de la corriente (La motocicleta puede ser encendida).

5.1.4 Diagramas de secuencia

5.1.4.1 Diagramas de secuencia de la aplicación móvil

5.1.4.1.1 Diagramas de secuencia - Registrar usuario

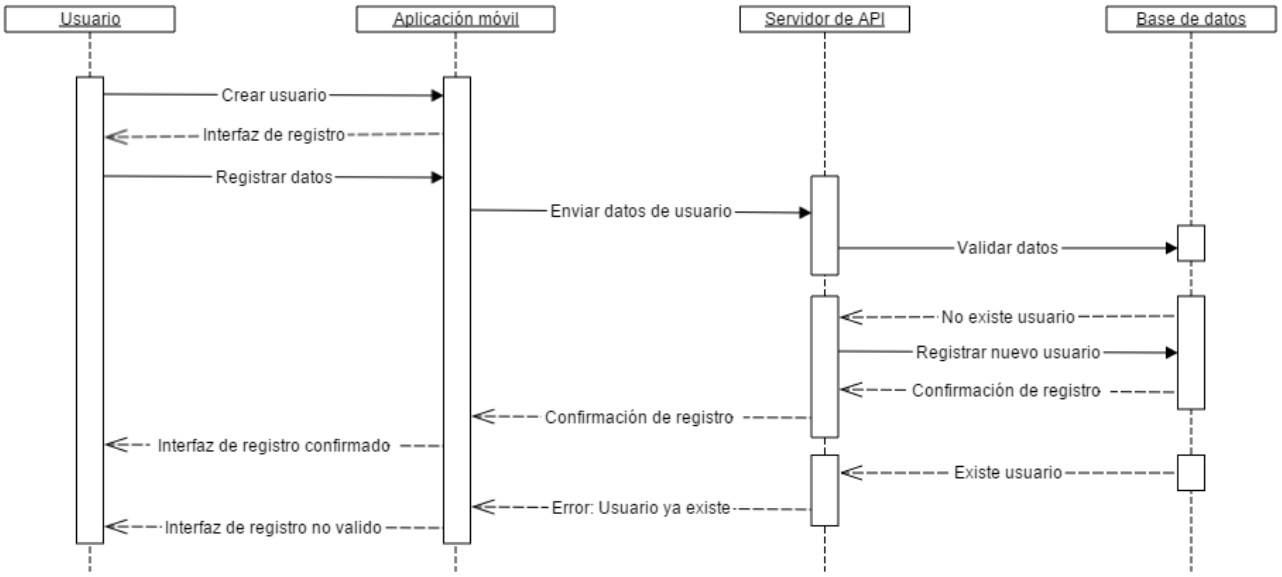

Figura 15: D.S – Registrar usuario en la aplicación móvil

5.1.4.1.2 Diagramas de secuencia – Iniciar sesión

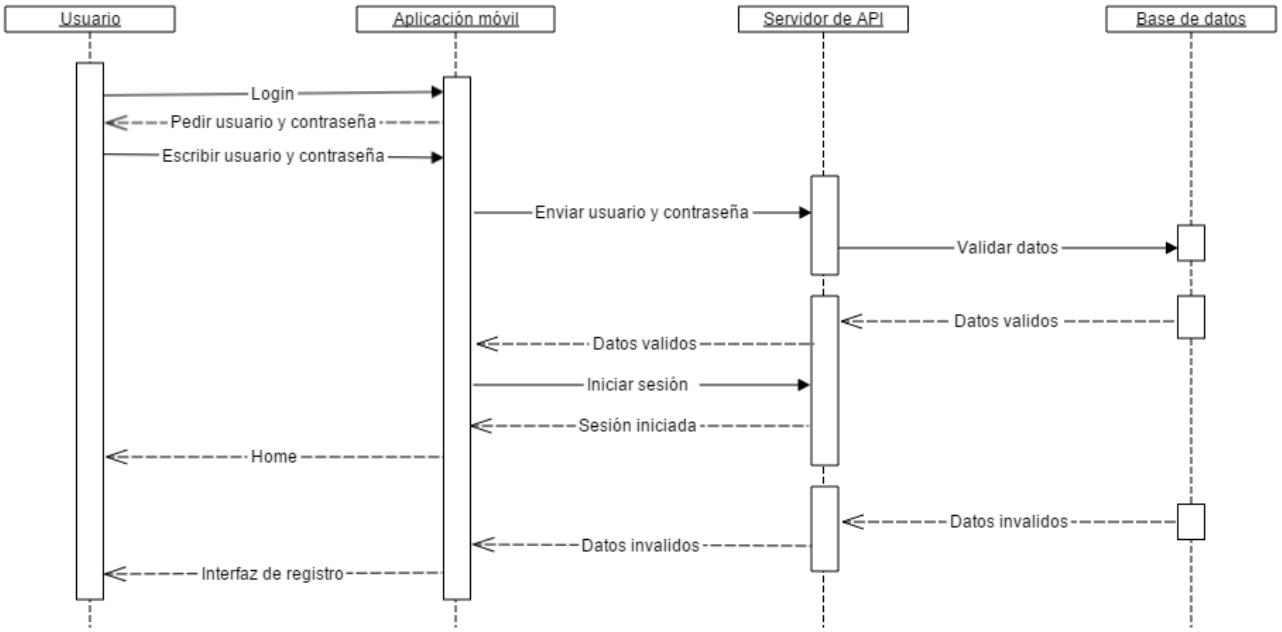

Figura 16: D.S – Iniciar sesión en la aplicación móvil

5.1.4.1.3 Diagramas de secuencia – Ver motocicletas

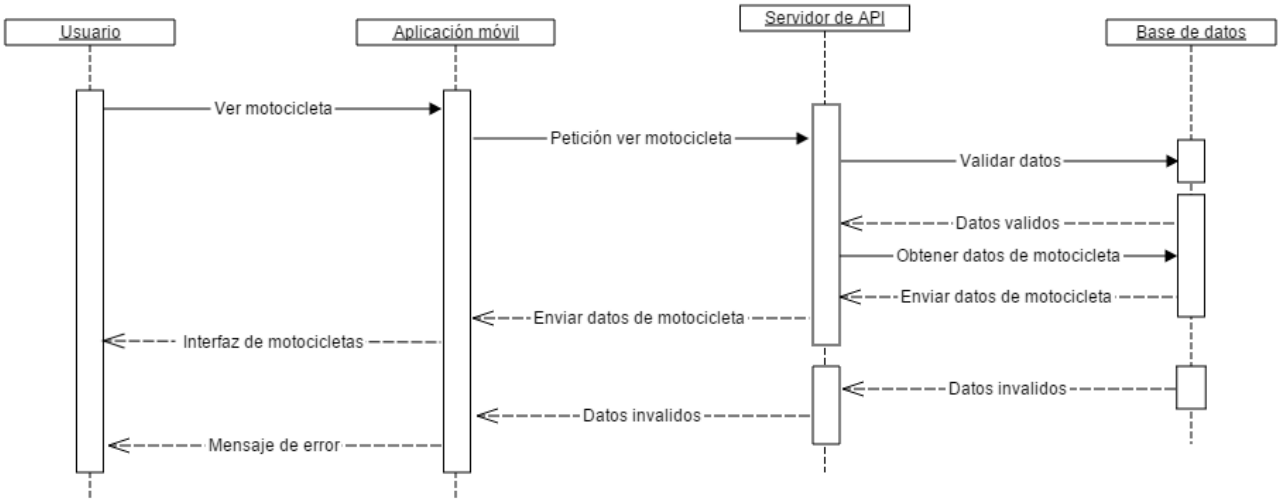

Figura 17: D.S – Ver motocicletas en la aplicación móvil

5.1.4.1.4 Diagramas de secuencia – Actualizar datos de motocicletas

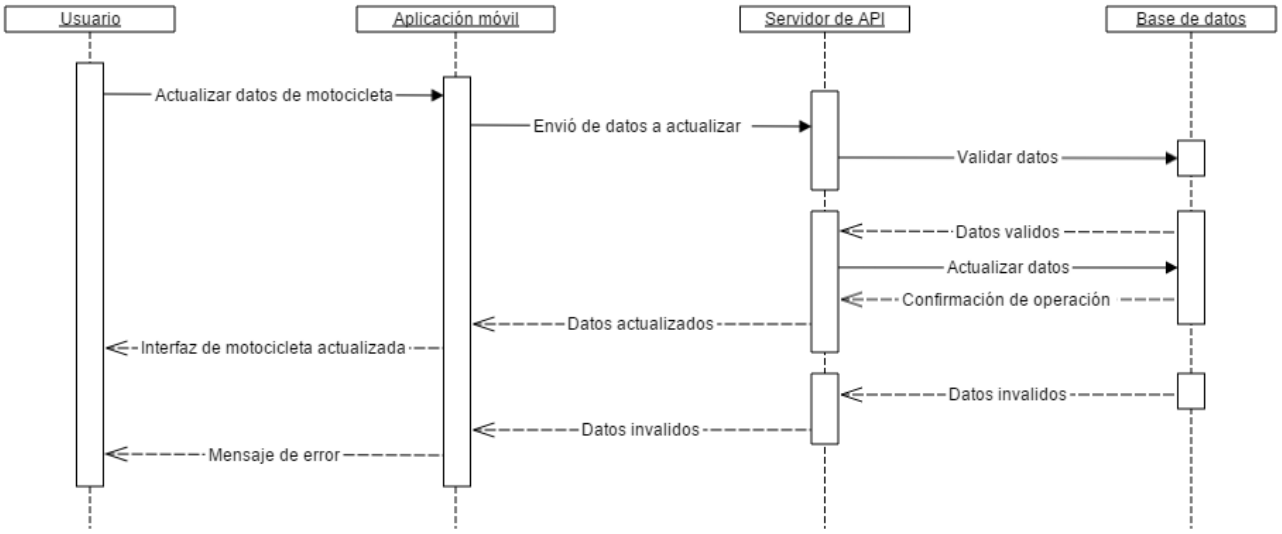

Figura 18: D.S – Actualizar motocicletas en la aplicación móvil

5.1.4.1.5 Diagramas de secuencia – Mostrar estado de los elementos monitoreados

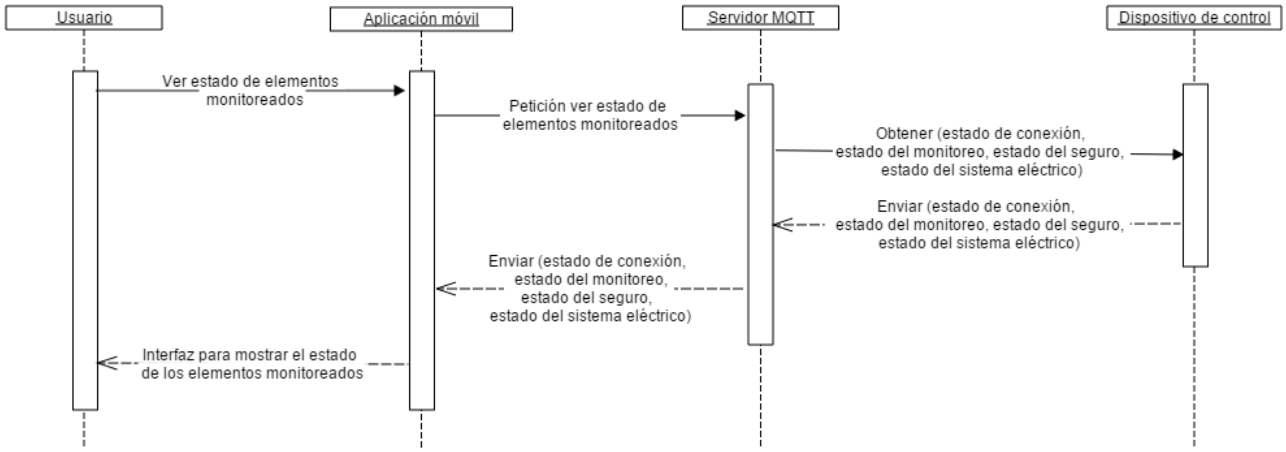

Figura 19: D.S – Mostrar estado de los elementos monitoreados

5.1.4.1.6 Diagramas de secuencia – Mostrar posición de la motocicleta

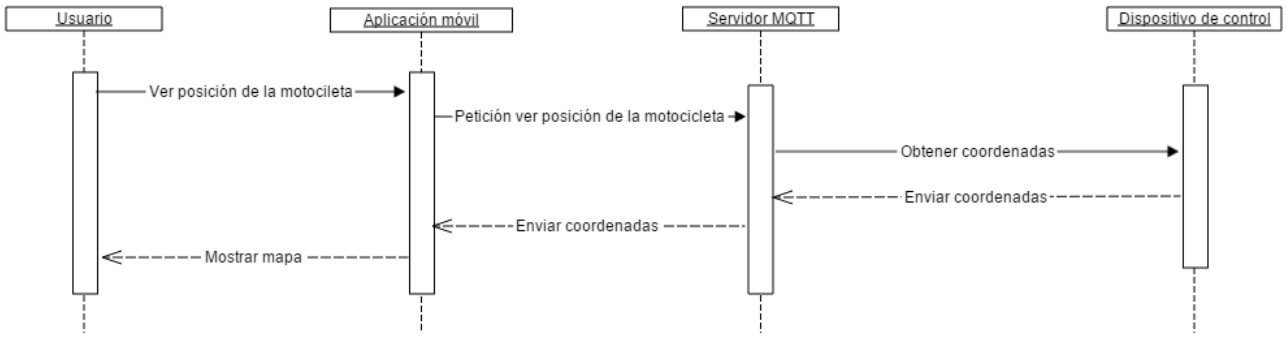

Figura 20: D.S – Mostrar posición de la motocicleta

5.1.4.1.7 Diagramas de secuencia – Activar monitoreo

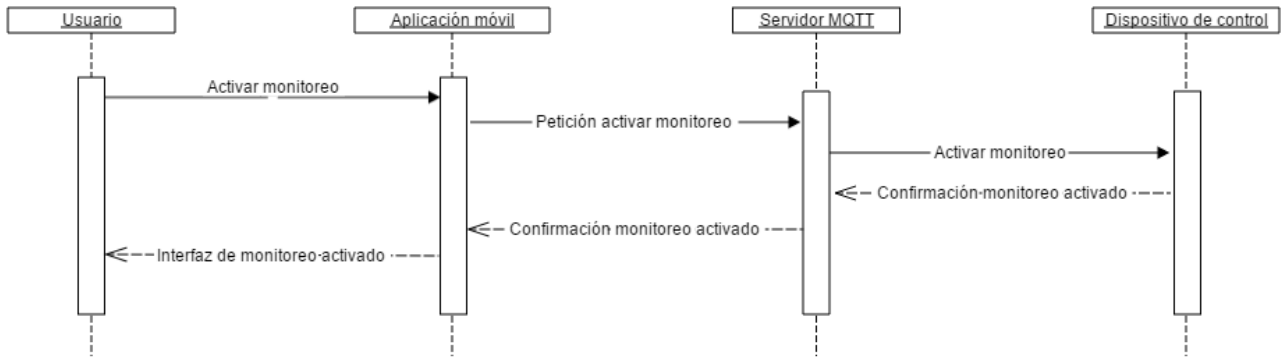

Figura 21: D.S – Activar monitoreo

5.1.4.1.8 Diagramas de secuencia – Notificar estado del seguro

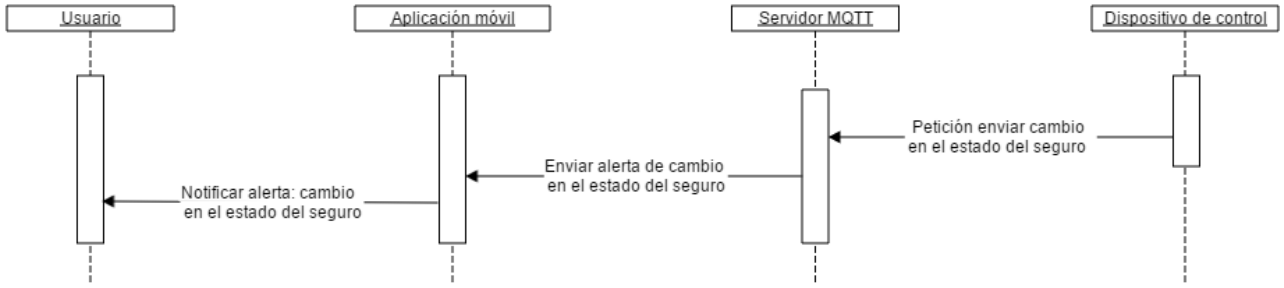

Figura 22: D.S – Notificar estado del seguro de la motocicleta

5.1.4.1.9 Diagramas de secuencia – Notificar desplazamientos

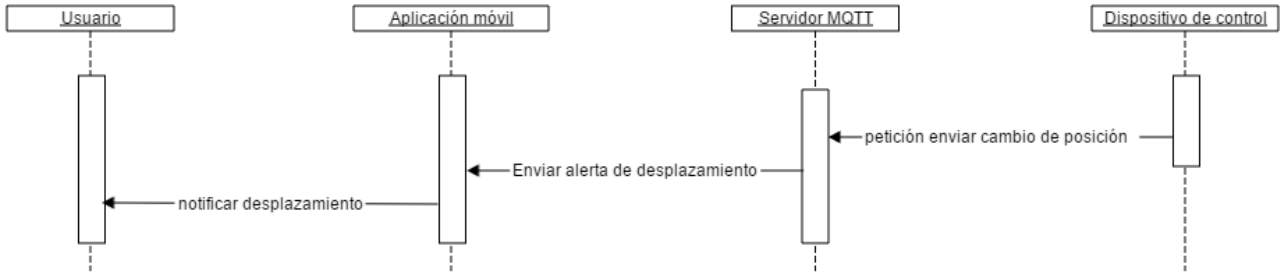

Figura 23: D.S – Notificar desplazamientos de la motocicleta

5.1.4.1.10 Diagramas de secuencia – Rastrear motocicleta

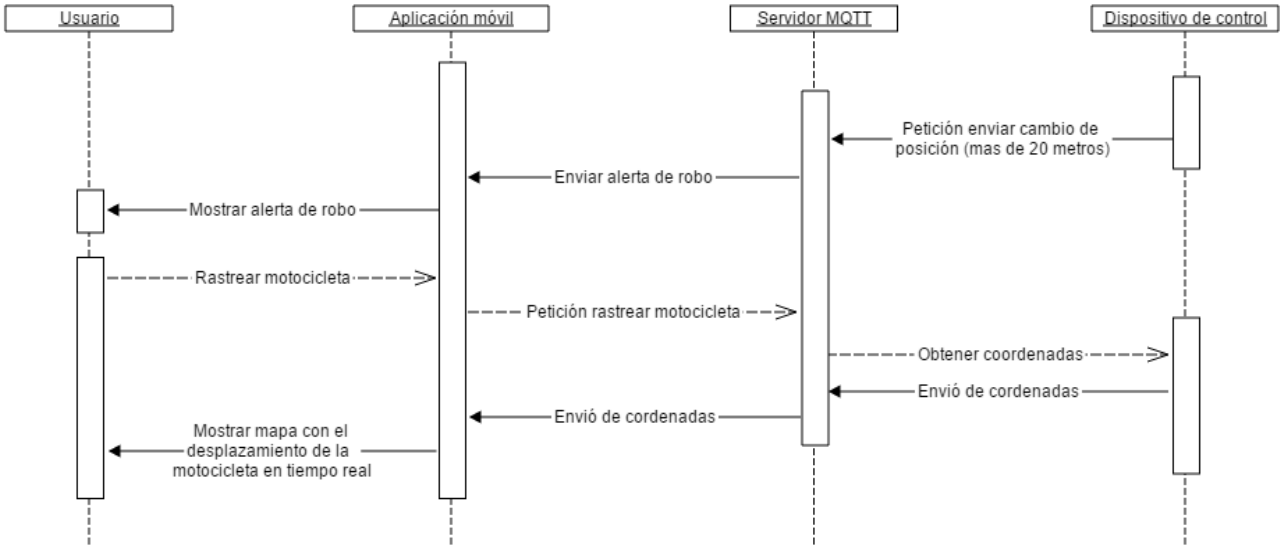

Figura 24: D.S – Rastrear motocicleta

5.1.4.1.11 Diagramas de secuencia – Desactivar monitoreo

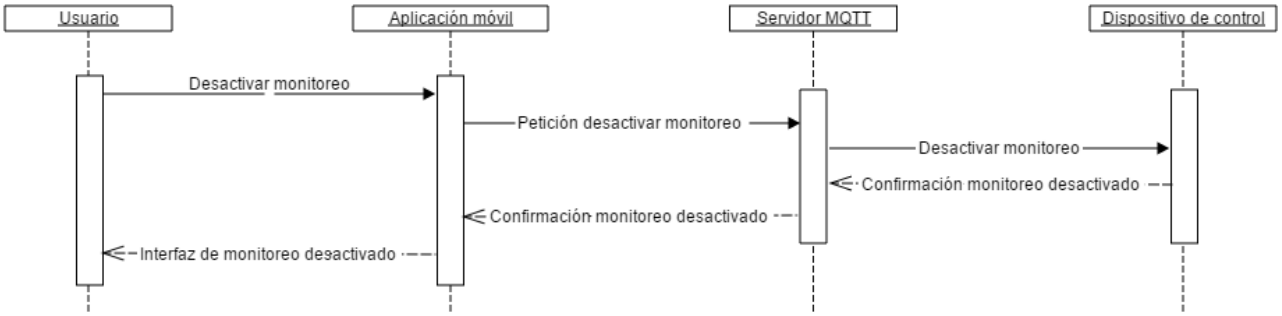

Figura 25: D.S – Desactivar monitoreo

5.1.4.1.12 Diagramas de secuencia – Cerrar sesión

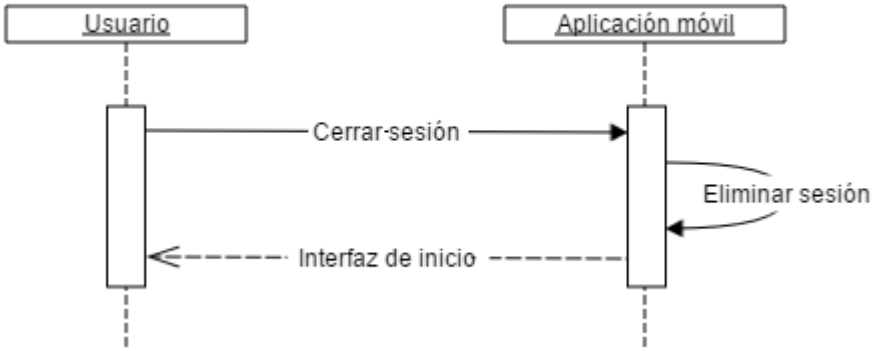

Figura 26: D.S – Cerrar sesión

5.1.4.2 Diagramas de secuencia de la aplicación web

5.1.4.2.1 Diagramas de secuencia – Registrar usuario

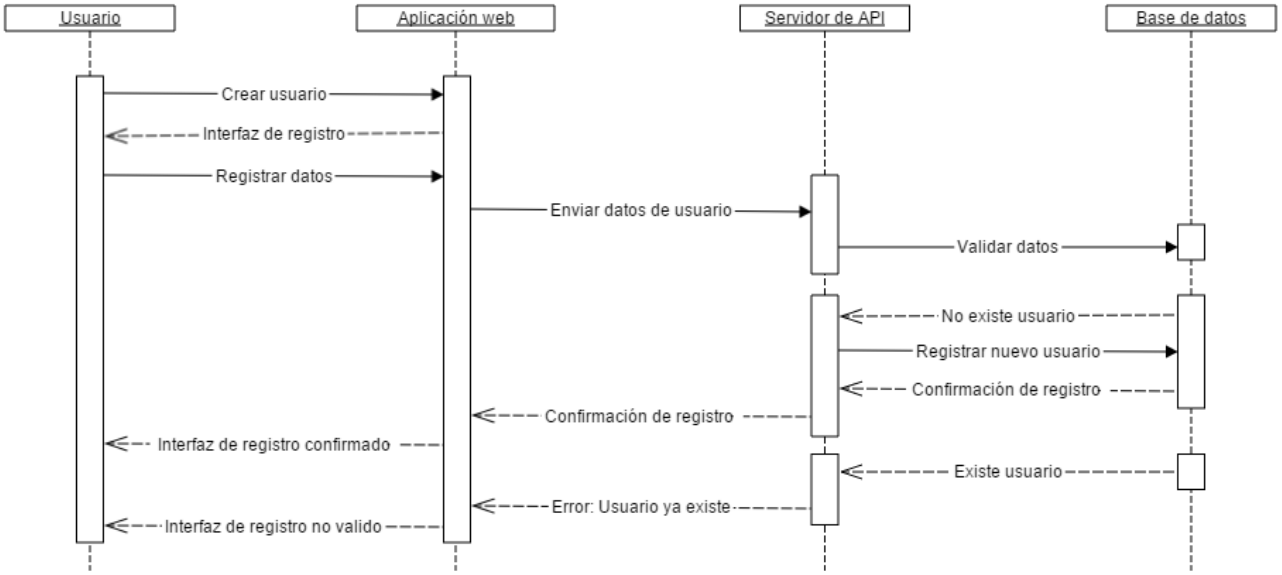

Figura 27: D.S – Registrar usuario en aplicación web

5.1.4.2.2 Diagramas de secuencia – Iniciar sesión

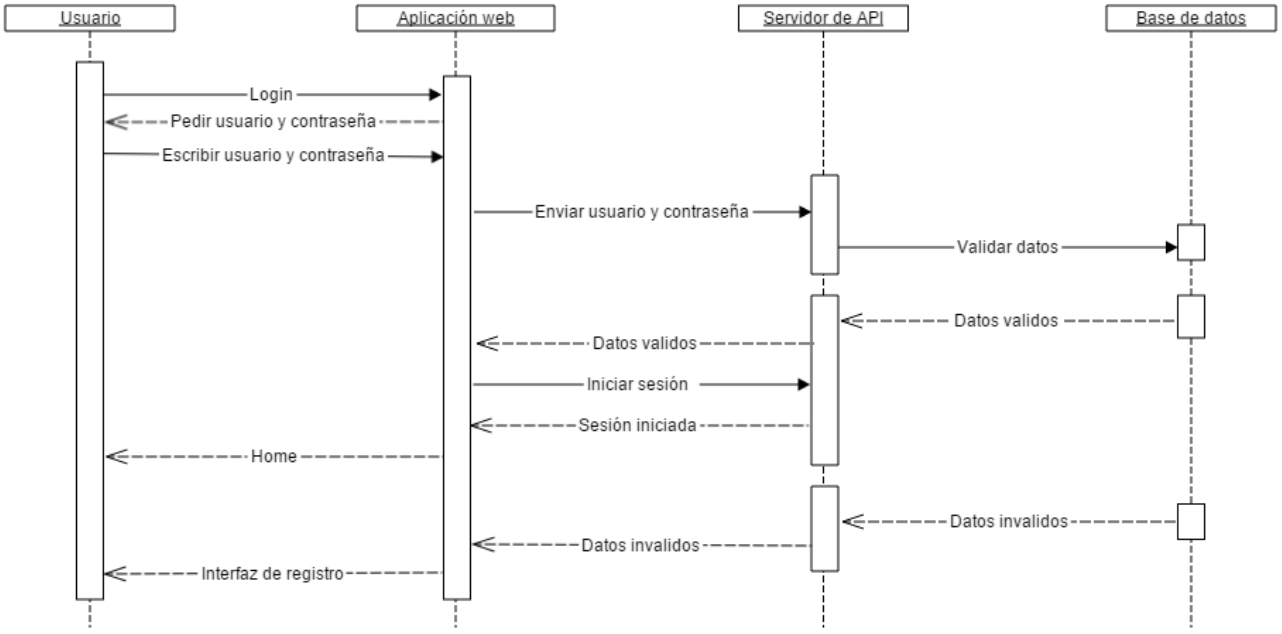

Figura 28: D.S – Inicio de sesión en la aplicación web

5.1.4.2.3 Diagramas de secuencia – Ver motocicletas

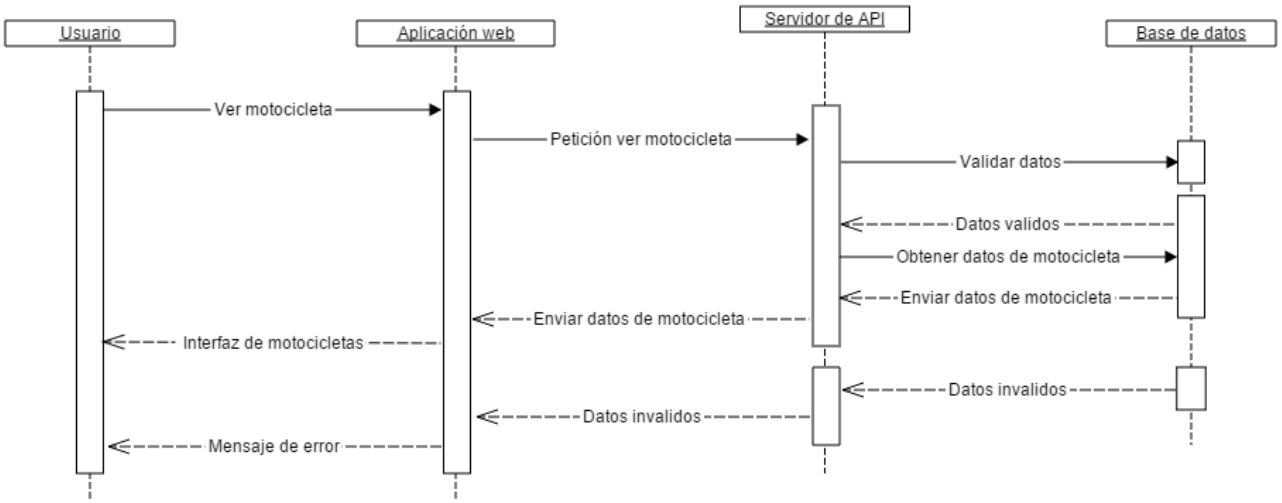

Figura 29: D.S – Ver motocicletas en la aplicación web

5.1.4.2.4 Diagramas de secuencia – Actualizar datos de motocicletas

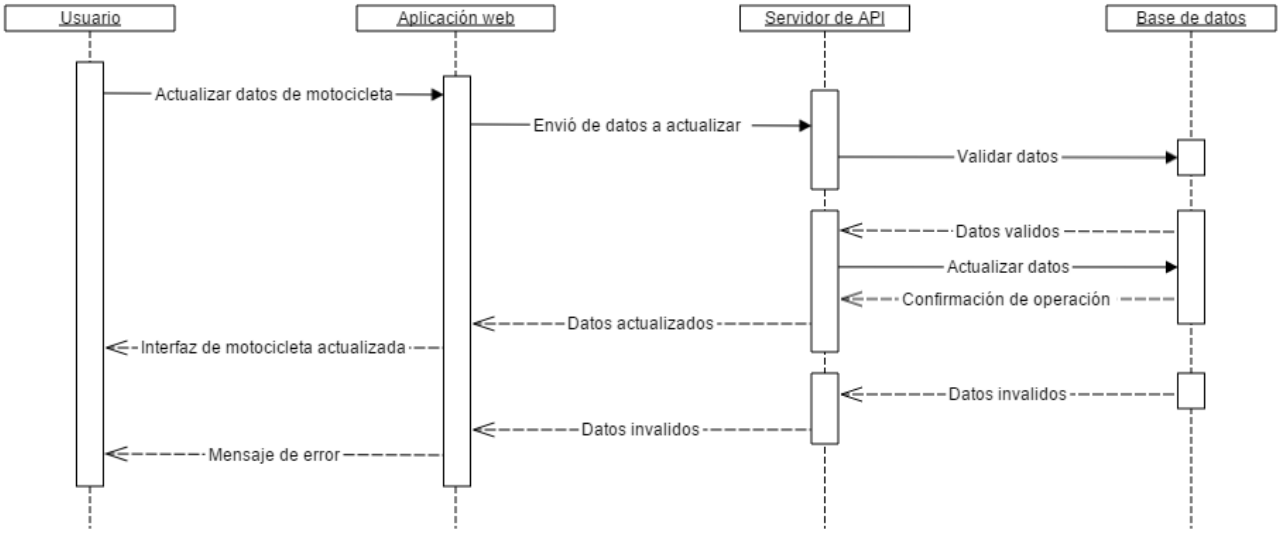

Figura 30: Actualizar datos de la motocicleta en la aplicación web

5.1.4.2.5 Diagramas de secuencia – Mostrar reportes de robo y recuperación de motocicletas

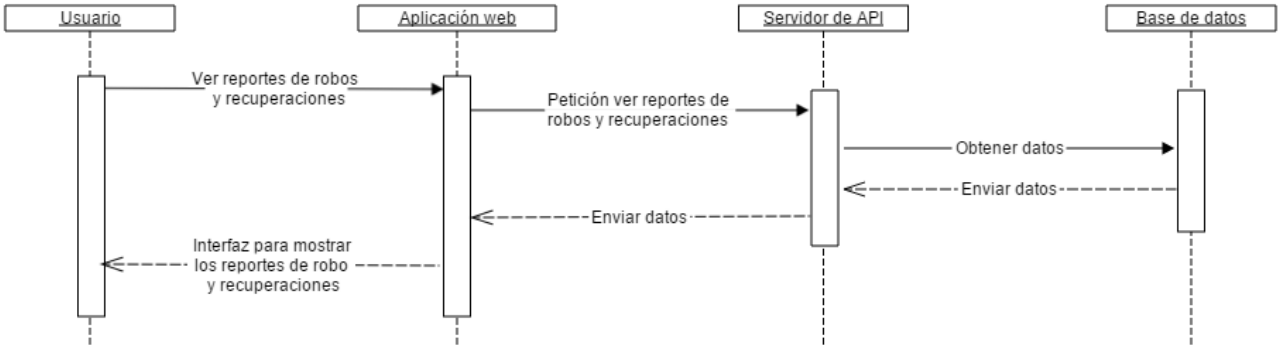

Figura 31: D.S – Mostrar reportes de robo y recuperación de motocicletas

5.1.4.2.6 Diagramas de secuencia – Cerrar sesión

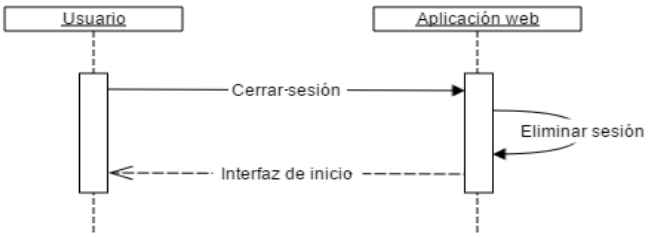

Figura 32: D.S – Cerrar sesión en la aplicación web

5.1.5 Diagrama de clases

A continuación se muestra el diagrama de clases de la aplicación móvil.

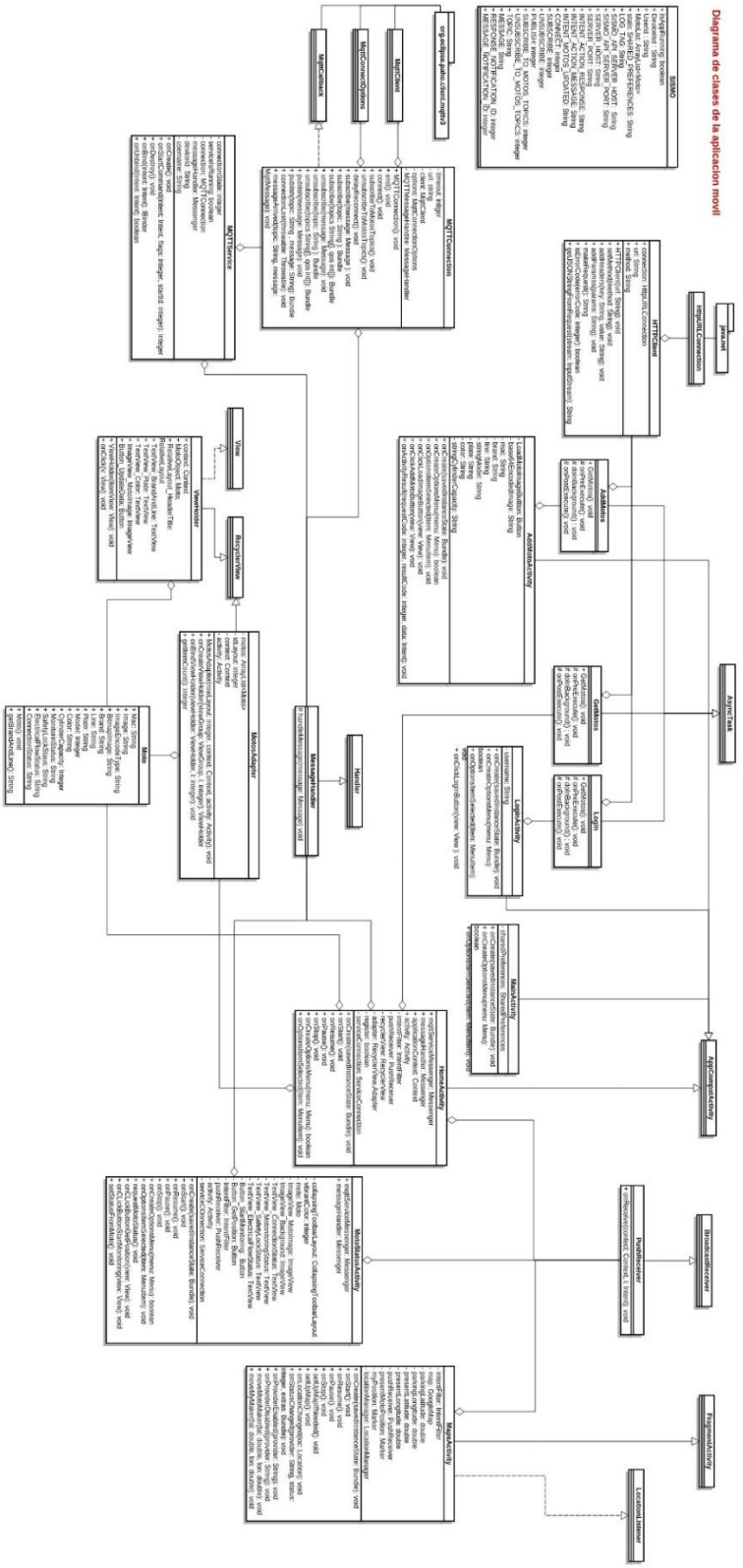

Figura 33: Diagrama de clases general de la aplicación móvil

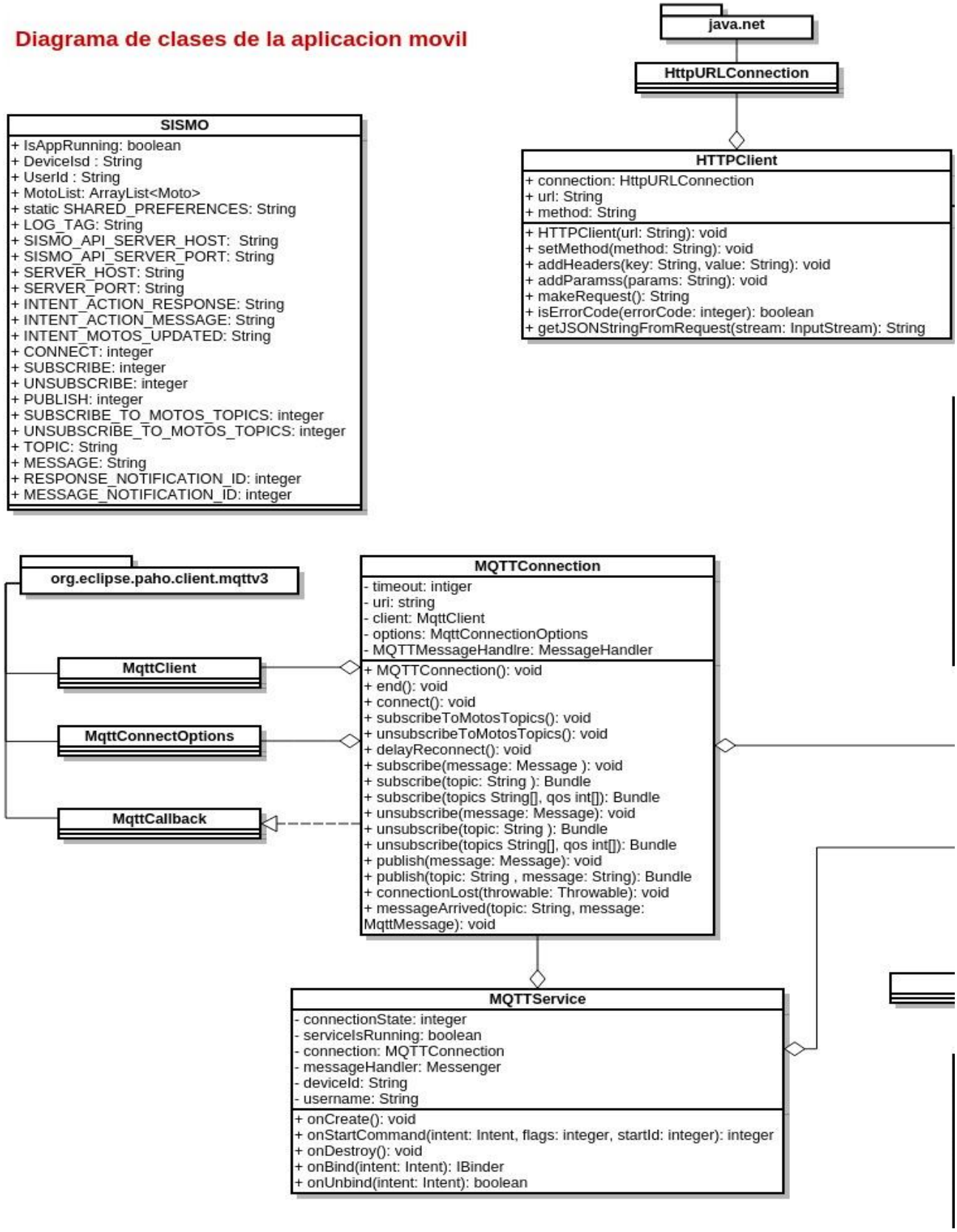

Figura 34: Diagrama de clases de la aplicación móvil - parte 1

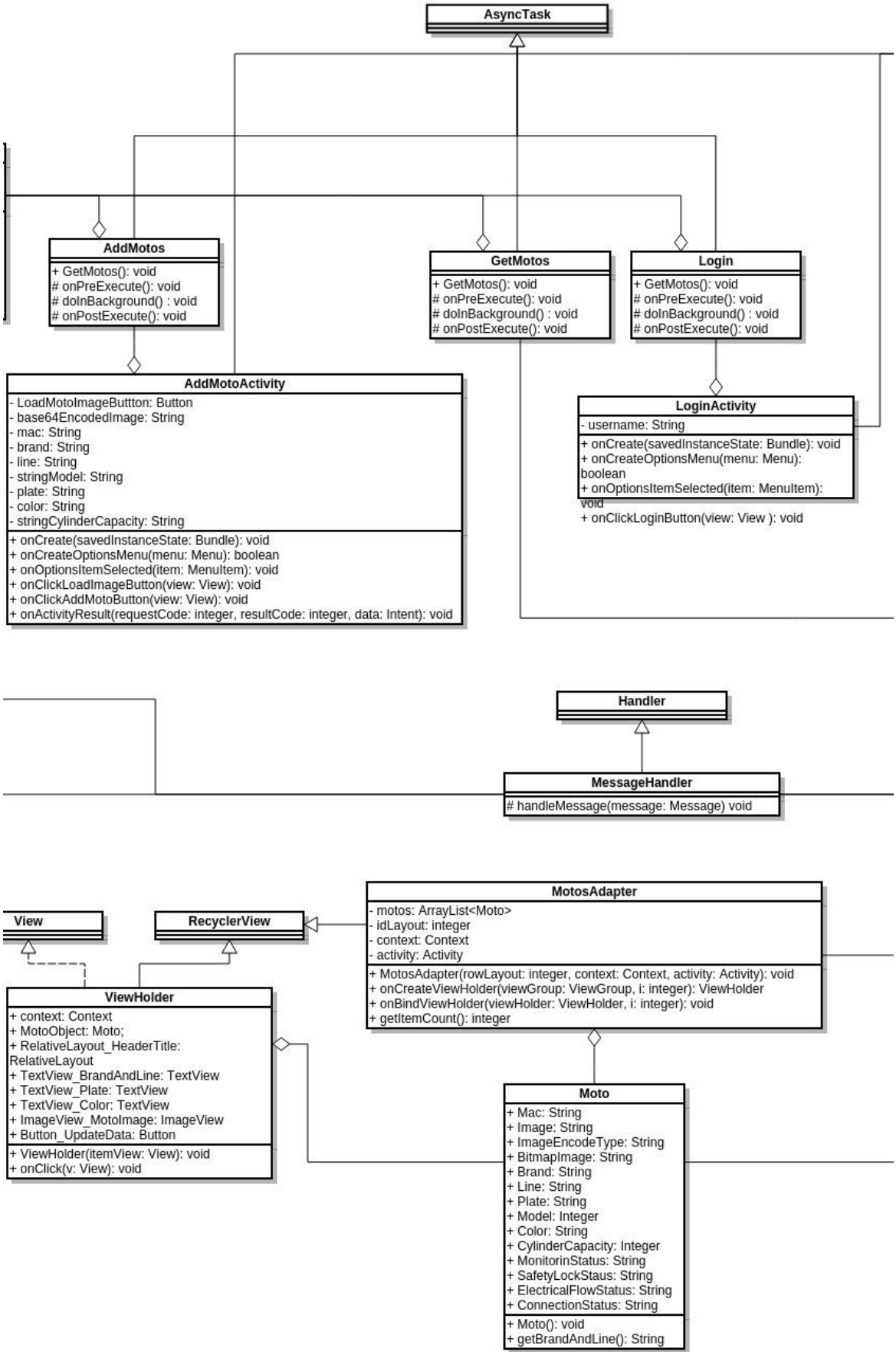

Figura 35: Diagrama de clases de la aplicación móvil - parte 2

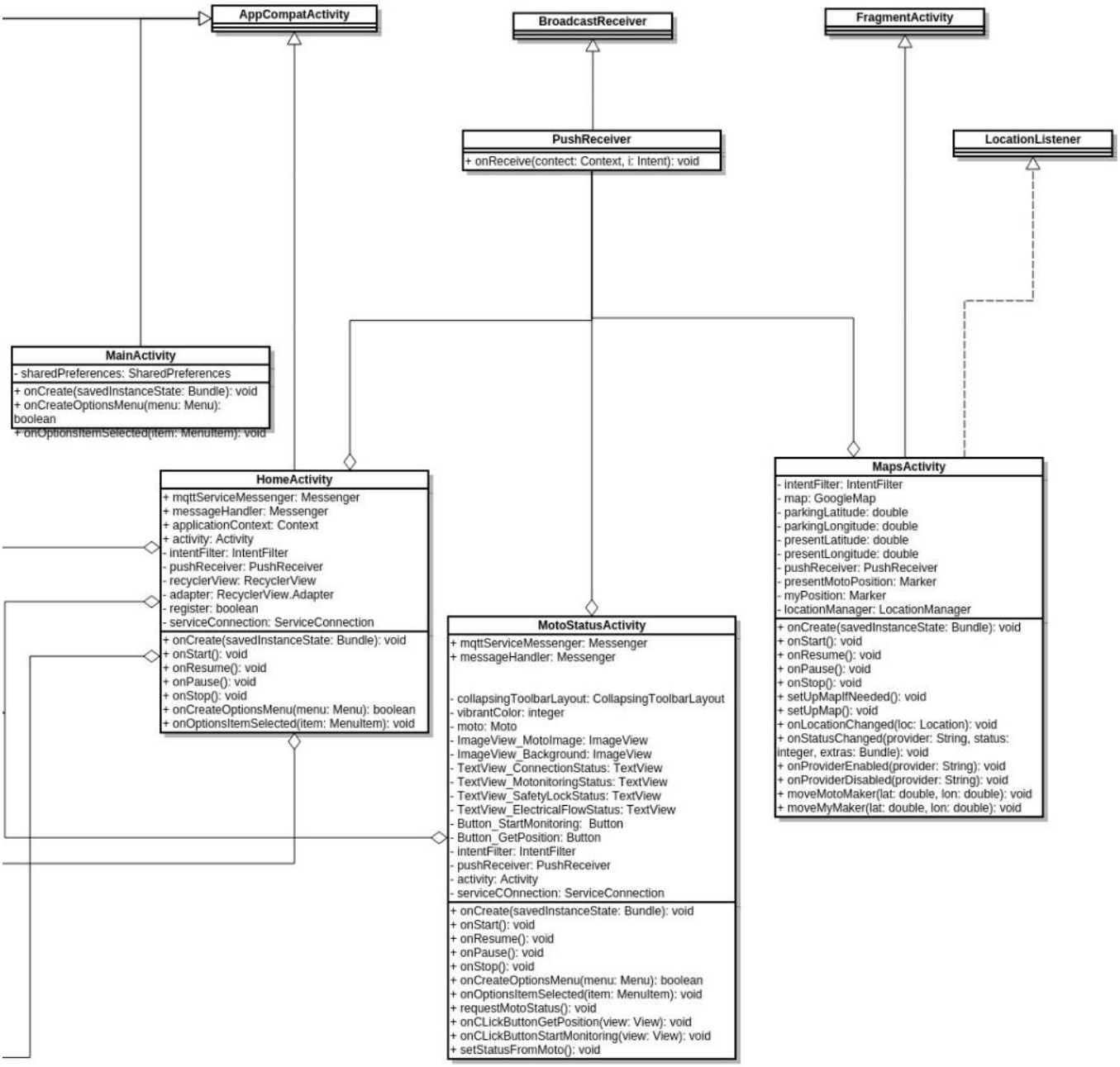

Figura 36: Diagrama de clases de la aplicación móvil - parte 3

5.2 Desarrollo del sistema

5.2.1 Desarrollo de la aplicación móvil

La aplicación móvil es una aplicación nativa para el sistema operativa Android, por lo tanto para el desarrollo de esta se utilizó el IDE oficial de Android llamado **Android Studio**, en este IDE se creó todo el código necesario para el completo funcionamiento de dicha aplicación móvil.

A continuación se puede observar una imagen del IDE mencionado anteriormente.

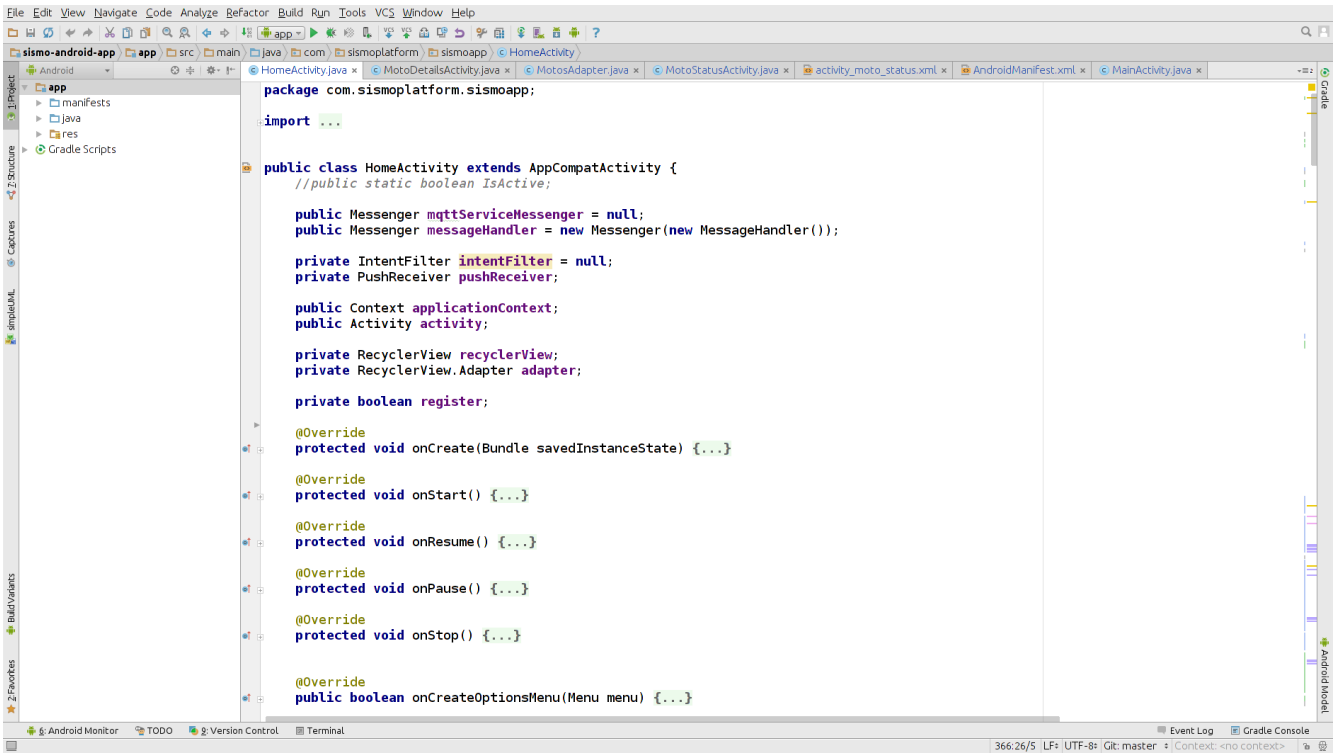

Figura 37: IDE Android Studio

Además de utilizar este IDE también cabe anotar el uso de una librería importante llamada **PAHO** creada por los desarrolladores del **IDE Eclipse** la cual permite a la aplicación hacer uso del protocolo **MQTT**, y sin la cual no se hubiese podido llevar a buen término el desarrollo de esta aplicación.

Sitio web de la librería **PAHO**: **http://www.eclipse.org/paho/**

A continuación también se puede observar algunas de las interfaces de la aplicación móvil.

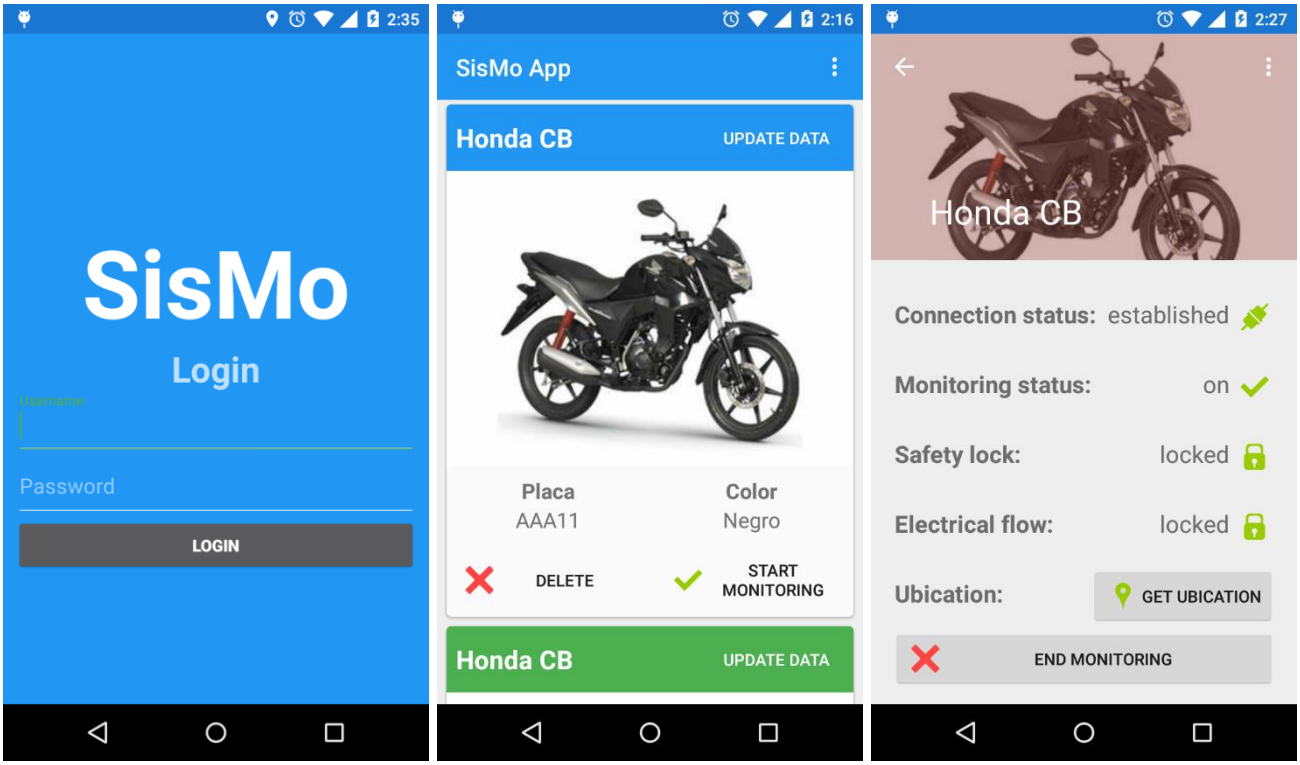

Figura 38: Interfaces aplicación móvil

5.2.2 Desarrollo del dispositivo electrónico

Para el desarrollo del dispositivo electrónico, se utilizó una placa Arduino, un módulo que le brinda conexión con el sistema y algunos componentes electrónicos para poder monitorear y controlar algunos componentes de la motocicleta.

El desarrollo de este dispositivo electrónico consta de 2 partes:

Construcción del dispositivo electrónico:

A continuación se pueden observar algunas imágenes de la construcción del dispositivo:

Figura 39: Componentes del dispositivo por separado

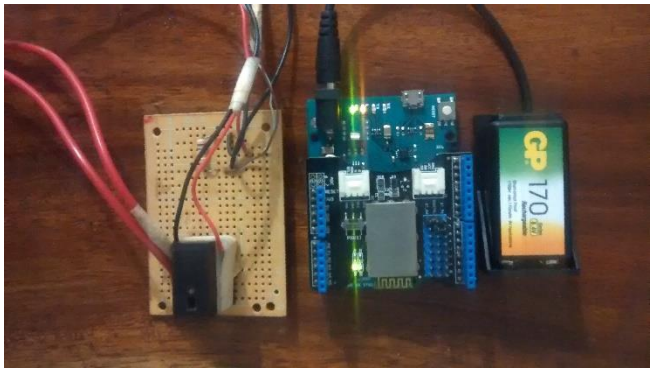

Figura 40: Dispositivo armado y conectado

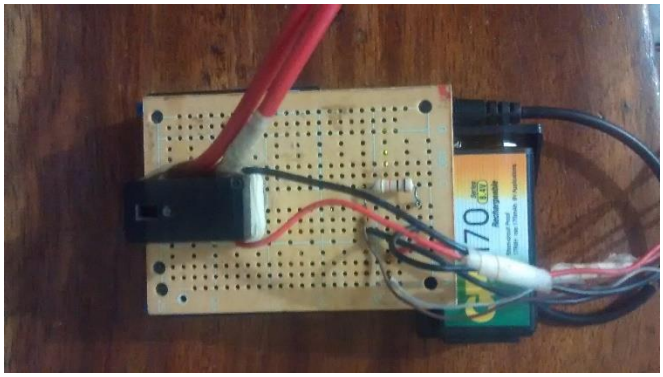

Figura 41: Dispositivo armado de la manera más compacta

Programación del dispositivo electrónico:

Para la programación del dispositivo se utilizó el IDE oficial de Arduino llamado **Arduino IDE** y se programó utilizando el lenguaje de programación **Processing** (Figura 39).

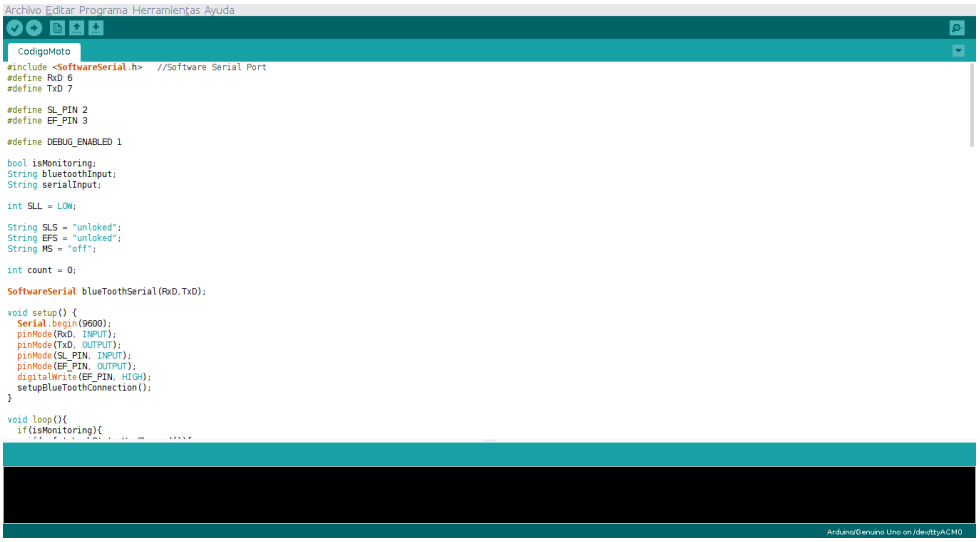

Figura 42: Arduino IDE y lenguaje de programación Processing

5.2.3 Desarrollo de la aplicación web

Para el desarrollo de la aplicación web se utilizaron tecnologías como **HTML5**, **CSS3** y **Javascript**, además el desarrollo se hizo utilizando un editor de texto llamado **Sublime Text 3** (Figura 43).

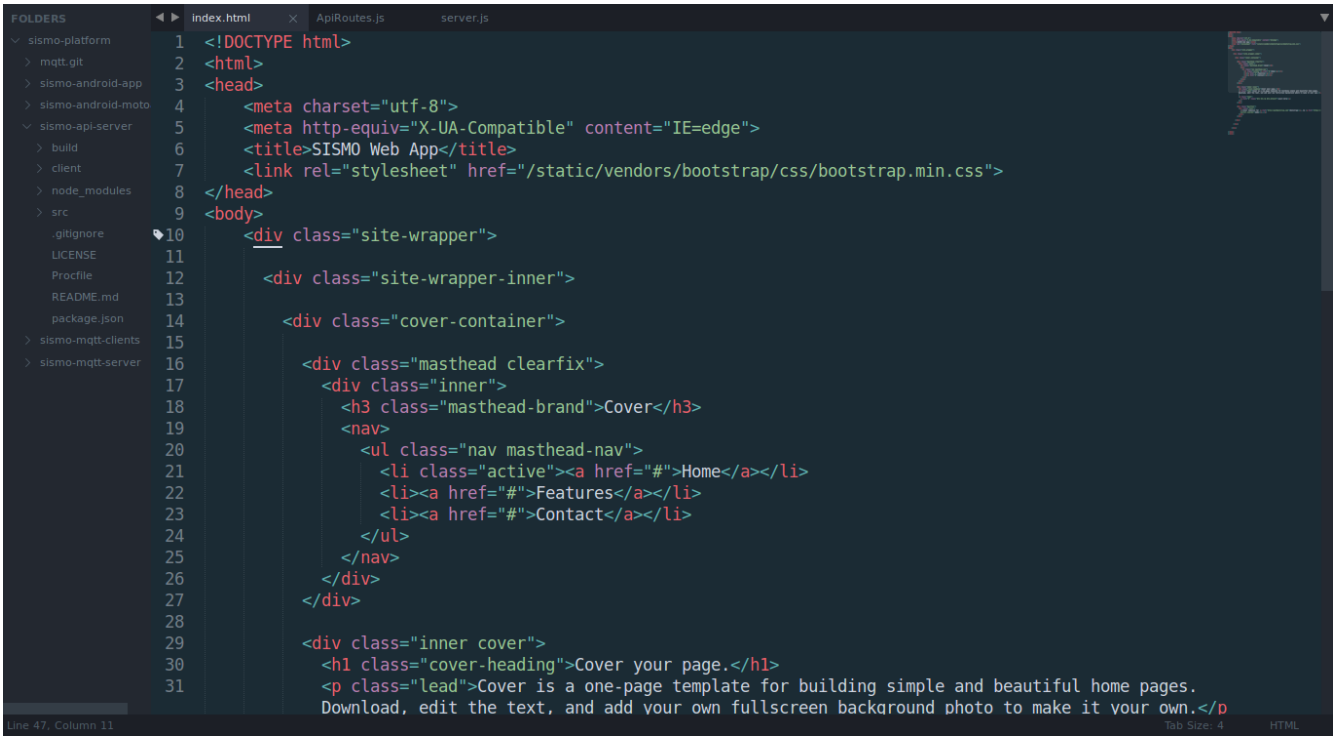

Figura 43: Desarrollo de la aplicación web en el editor de texto Sublime Text 3

5.2.4 Desarrollo del servidor API REST

Las tecnologías utilizadas para desarrollar el servidor de la API REST fueron:

- **Node.js**: Servidor web.
- **MongoDB**: Motor de base de datos.
- **Express.js**: Framework de desarrollo.
- **Javascript versión 6 (ES6)**: Lenguaje de programación.

Para poder desarrollar el servidor de la API REST con la versión 6 de JavaScript se utilizó una librería para **transpilar** el código de ES6 a ES5 llamada **Babel**, esta librería nos permite utilizar todas las funcionalidades nuevas de Javascript (ES6) que aun Node.js no soporta nativamente.

Además el servidor se desarrolló utilizando el editor **Sublime Text 3** (Figura 44) y se probaba de manera local mediante la consola (Figura 45).

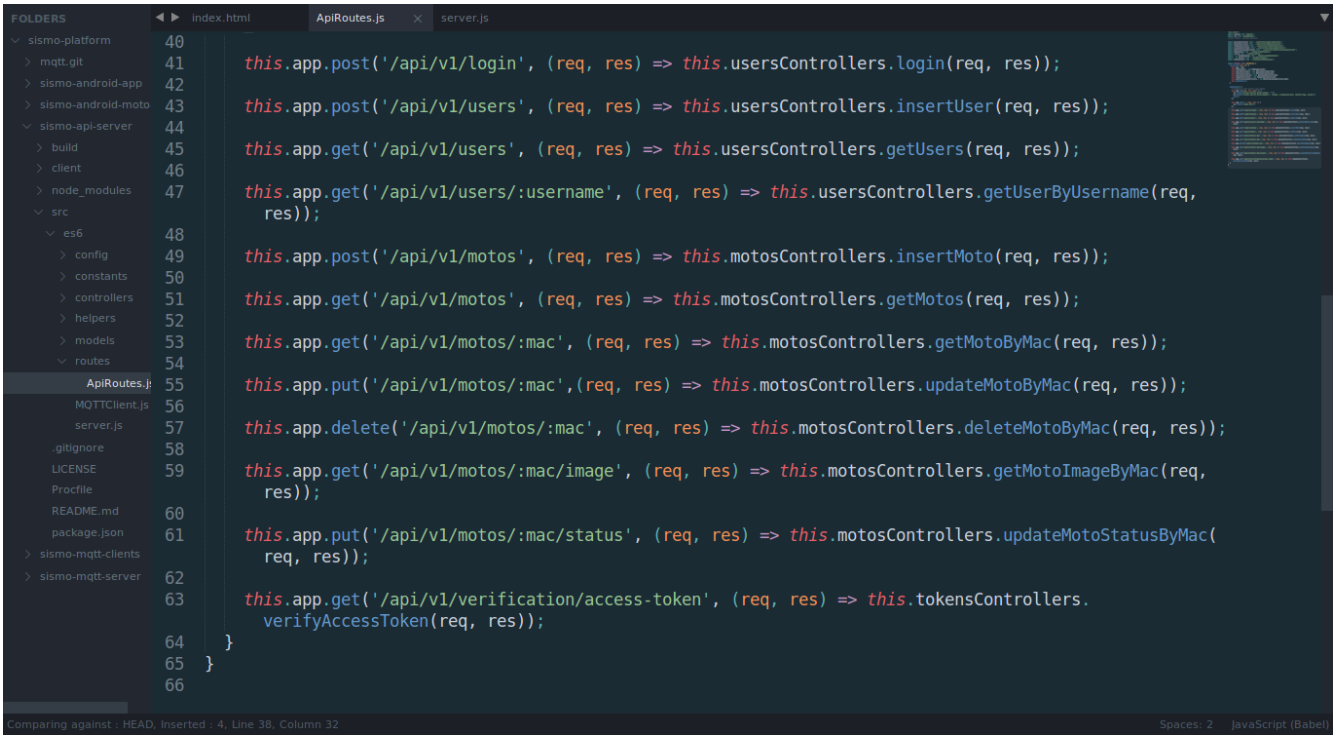

Figura 44: Desarrollo de la API REST en el editor de texto Sublime Text 3

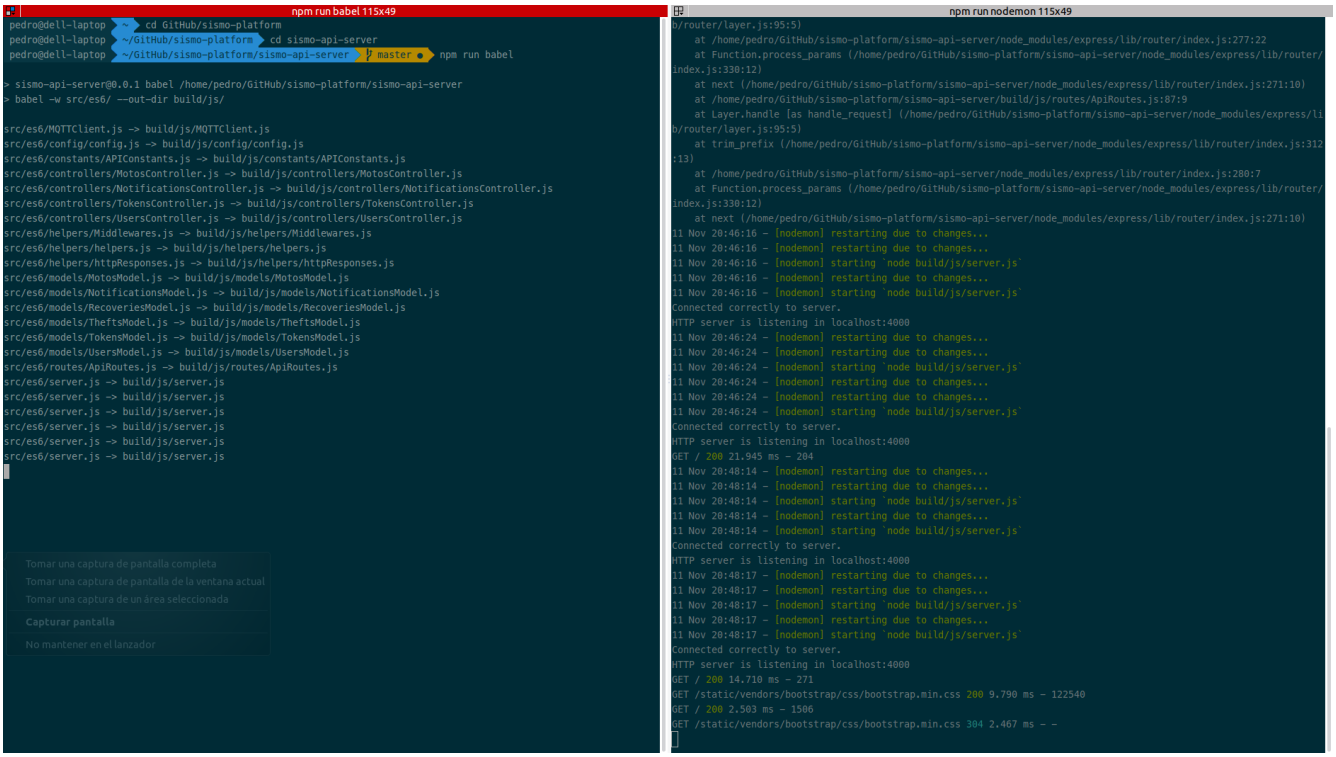

Figura 45: Ejecución de la API REST en la consola

5.2.5 Desarrollo de la base de datos

El motor de base de datos seleccionado fue **MongoDB**, donde se crearon los siguientes esquemas:

```
users = {
  id: objectId,
  account: {
    username: string,
    password: string
  }
}
```

Figura 46: Esquema para usuarios

```
motos = {
  id: ObjectId,
  userId: OnjectId,
  mac: string,
  brand: string,
  line: string,
  model: int,
  plate: string,
  color: string,
  cylinderCapacity: int,
  image: string,
  imageEncodeType: string,
  status: {
    monitoring: string,
    electricalFlow: string,
    safetyLock: string,
    parkingLatitude: double,
    parkingLatitude: double
  }
}
```

Figura 47: Esquema para motocicletas

```
thefts = {
  id: ObjectId,
  userId: ObjectId,
  moto: {
    mac: string,
    brand: string,
    line: string,
    model: int,
    plate: string,
    color: string,
    cylinderCapacity: int
  },
  location: {
    latitude: string,
    longitude: string,
    country: string,
    department: string,
    city: string,
    address: string,
  },
  date: {
    year: int,
    month: int,
    day: int
  }
}
```

Figura 48: Esquema para motocicletas robadas

```
recoveries = {
  id: ObjectId,
  theftId: ObjectId,
  location: {
    country: string,
    department: string,
    city: string,
    address: string,
  },
  date: {
    year: int,
    month: int,
    day: int
  }
}
```

Figura 49: Esquema para motocicletas recuperadas

6. PRUEBAS Y RESULTADOS

En el desarrollo e implementación del sistema de seguridad (**SisMo**), se realizaron una serie de pruebas para verificar el correcto funcionamiento de este y para encontrar errores que posteriormente fueron solucionados.

- **Prueba 1 (Conexión a internet):** Se verifico la transmisión y recepción de datos entre los servidores, la aplicación móvil y el dispositivo electrónico.

  Resultado: Luego de lograr la sincronización todos los datos fueron enviados y recibidos satisfactoriamente.

- **Prueba 2 (Conexión dispositivo/aplicación móvil**): Se verifico la transmisión y recepción de datos utilizando el protocolo MQTT entre el dispositivo electrónico y la aplicación móvil.

  Resultado: Luego de establecer la comunicación se pudo observar que la comunicación fue rápida y sin ninguna interferencia.

- **Prueba 3 (seguro de la motocicleta):** Esta prueba se dividió en dos partes:

  - La primera consistió en pedirle a aplicación móvil que nos mostrara el estado de los elementos monitoreados antes de activar el monitoreo y sin ponerle el seguro a la motocicleta, esto se hizo con el fin de saber si el sistema era capaz de mostrarle al usuario el estado actual del seguro de su motocicleta. Lo mismo se hizo con el seguro de la motocicleta ya puesto.

    Resultado: Sin haber activado el monitoreo el sistema fue capaz de reconocer el estado del seguro cuando estaba sin poner y cuando fue puesto.

  - Para la segunda parte de la prueba, se le puso el seguro a la motocicleta y se activó el monitoreo del sistema, luego, con el monitoreo activado se le quito el seguro a la motocicleta con el fin de verificar si la alerta correspondiente era recibida por el usuario. Por último se volvió a ponerle el seguro a la motocicleta estando el monitoreo activo.

Resultado: Al quitar el seguro de la motocicleta estando activado el monitoreo, inmediatamente el sistema mostro la alerta correspondiente. Lo mismo ocurrió cuando se volvió a poner el seguro.

- **Prueba 4 (Prueba del encendido):** Se encendió la motocicleta, luego se activó el monitoreo del sistema con fin de verificar que la motocicleta se apagara automáticamente y que se bloqueara el encendido de esta. Luego se realizó lo contrario, se desactivo el monitoreo del sistema para verificar que el encendido de la motocicleta se desbloqueará.

  Resultado: Al activar el monitoreo estando la motocicleta encendida, inmediatamente esta se apagó y no se pudo volver a encenderla ni por encendido electrónico ni por la palanca de encendido (crank) hasta que no se desactivo el monitoreo.

- **Prueba 5 (Prueba de desplazamiento):** Esta prueba se dividió en tres partes:

  - La primera consistió en parquear la motocicleta en un punto fijo y pedirle a aplicación móvil que nos mostrara la posición de la misma antes de activar el monitoreo, esto se hizo con el fin de comprobar si el sistema era capaz de mostrarle al usuario el mapa con la posición de su motocicleta aun sin ser activado el monitoreo.

    Resultado: Sin haber activado el monitoreo el sistema fue capaz de mostrarle al usuario la posición de su motocicleta a través de Google Maps.

  - Para la segunda parte de la prueba, se activó el monitoreo del sistema, luego, se empezó a mover la motocicleta desde el punto inicial donde estaba estacionada, con el fin de verificar que llegaran al móvil del usuario las respectivas alertas de desplazamiento de 5, 15 y 25 metros con respecto a la posición de parqueo.

    Resultado: Al mover la motocicleta estando activado el monitoreo, inmediatamente el sistema mostró las respectivas alertas a los: 5, 15 y 25 metros con respecto al punto de referencia.

  - En la última parte de la prueba se realizaron los mismos pasos del punto

anterior, pero teniendo en cuenta de que cuando la motocicleta pasara de los 25 metros de distancia del punto de referencia, se lanzara en la aplicación del usuario la alerta para mostrar el mapa de Google Maps, para permitir rastrear la motocicleta en tiempo real. También se verifico en esta prueba que la posición de la motocicleta mostrada en los mapas fuera lo más precisa posible.

Resultado: Al mover la moto y sobrepasar los 25 metros de distancia del punto de referencia inmediatamente llego la alerta para el rastreo de la motocicleta y se pudo ubicar con exactitud y en tiempo real la ubicación de la misma.

Todas las notificaciones y alertas enviadas a la aplicación móvil del usuario tienen un tiempo de respuesta de: 3 Segundos mejores casos, 6 segundos casos promedio y 10 segundos peores casos.

7. CONCLUSIONES

Las investigaciones revisadas a lo largo de este proyecto dejan en evidencia la gran problemática que enfrenta el país en cuanto a hurto de motocicletas, por esta razón la policía nacional en conjunto con otras entidades gubernamentales han lanzado campañas que buscan desincentivar el hurto de estos vehículos, cabe aclarar que estas campañas son más que todo educativas y enfocadas a prevenir este hecho delictivo dejando de lado la seguridad como tal en estos vehículos y la capacidad de recuperar inmediatamente las motocicletas en caso de ser hurtadas. También se encontró que en los últimos años se han venido desarrollando dispositivos de seguridad para motocicletas que hacen uso de nuevas tecnologías como el GPS y controles remotos, pero brindan funciones específicas y aisladas para garantizar la seguridad de las motocicletas. Este proyecto busca integrar las mejores funcionalidades encontradas en dichos dispositivos en un sistema completo de seguridad para motocicletas, cubriendo las necesidades del usuario respecto a la prevención y recuperación de sus vehículos.

En la actualidad la demanda de dispositivos móviles tiene un gran auge, estos dispositivos son usados por niños, jóvenes y adultos, la demanda de estos dispositivos es tan grande que las empresas están apuntando a ofrecer sus productos y servicios a través de estos para así abarcar un mayor mercado de consumidores, teniendo en cuenta esto se puede decir que los productos o servicios que no se ofrezcan a través de estas tecnologías serán desplazados por aquellos que si lo hagan. Por esta razón se ha desarrollado nativamente una aplicación para el sistema operativo Android, el cual es el de mayor demanda dentro del mercado de dispositivos móviles, esto, más la posibilidad de poderla descargar de manera gratuita, facilitara la posibilidad de llegar a más personas logrando una mayor cobertura de mercado. Esta aplicación es una parte fundamental del módulo de software que compone al sistema, que además de este, también cuenta con un módulo de hardware (Dispositivo electrónico) que va integrado dentro de la motocicleta.

Por último se debe tener en cuenta que las tecnologías por si solas no resuelven todos los problemas, pero si se logra una implementación sistemática y coordinada de estas, pueden ayudar a ir por el camino correcto. Lo importante no es tener la tecnología, si no utilizarla de la mejor forma enfocándola a un problema real y concreto.

REFERENCIAS BIBLIOGRÁFICAS

ALARCOM. (2012). GPS ubicación satelital, alarma gsm para motos. Recuperado el 21 de 11 2013, de http://www.youtube.com/watch?v=i4HQctID27A

Alejandra. (2007). elWebmaster. API, Interface de Programación de Aplicaciones. Recuperado el 18 de 05 2014 de http://www.elwebmaster.com/referencia/api-interface-de-programacion-de- aplicaciones

Alejandro L. Veiga. (S.F). Sistemas de tiempo real. Recuperado el 05 de 11 de 2015, de http://www.electro.fisica.unlp.edu.ar/temas/p7/RTS-1.html

Alejandro Nieto González. (2011). ¿Qué es Android?. Recuperado el 16 de 05 de 2014, de http://www.xatakandroid.com/sistema-operativo/que-es-android

Arduino. (2015). Arduino Leonardo. Recuperado el 05 de 11 de 2015, de https://www.arduino.cc/en/Main/ArduinoBoardLeonardo

Asier Marqués. (2013). Conceptos sobre APIs REST. Recuperado el 03 de 11 de 2015, de http://asiermarques.com/2013/conceptos-sobre-apis-rest/

Asomocol. (2013). VIII Estudio Sociodemográfico del Comité de Ensambladoras Japonesas. Recuperado el 13 de 11 de 2013, de http://www.asomocol.org/noticias/viii-estudio-sociodemografico-del-comite-de-ensambladoras-japonesas

Bidcom. (2012). Localizador de gps GPS CARGO TRACKER. Recuperado el 21 de 11 de 2013, de http://www.bidcom.com.ar/gps-cargo-tracker-gpssmsgprs

CEil. (S.F). Técnicas y procesos de instalaciones singulares – Sistemas de seguridad.

Damián Pérez Valdés. 26 de 10 de 2007. Maestros del web. ¿Que son las bases de datos?. Recuperado el 26 de mayo de 2014, de http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/

Daniel Glez. 28 de 09 de 2015. 20 minutos. Objetos inteligentes: cuando la domótica evolucionó hacia el Internet de las cosas. Recuperado el 01 de 11 de 2015, de http://www.20minutos.es/noticia/2560747/0/domotica/internet-de-las-cosas/futuro/

Dave Evans. (2011). Internet de las cosas; Cómo la próxima evolución de internet lo cambia todo.

Diario ADN. (2012). Buscan disminuir el robo de motos en Cali. Diarioadn.co. Recuperado el

13 de 11 de 2013, de http://diarioadn.co/cali/mi-ciudad/buscan-disminuir-el-robo-de-motos-1.32962

EcuRed. (2014, 26 de mayo). Aplicación web. Fecha de consulta: mayo 26, 2014 desde http://www.ecured.cu/index.php/Aplicaci%C3%B3n_web

El meridiano de Córdoba. (2013). Robo de motos se disparó. Recuperado el 21 de 11 de 2013, de http://www.elmeridianodecordoba.com.co/index.php?option=com_k2&view=item&id=33537:robo-de-motos-se-dispar%C3%B3&Itemid=114

El nuevo siglo. (13 de 07 de 2014). Elnuevosiglo.co. Recuperado el 22 de 09 de 2014, de http://www.elnuevosiglo.com.co/articulos/7-2014-cada-hora-en-el-pa%C3%ADs-se-roban-dos-motocicletas.html

El Tiempo. (S.F). ¿Qué es el internet de las cosas?. Eltiempo.com. Recuperado el 01 de 11 de 2015, de: http://www.eltiempo.com/multimedia/infografias/que-es-el-internet-de-las-cosas/15119081

El Tiempo. (2015). Todos los días se roban 64 motos en ciudades capitales de Colombia. Eltiempo.com. Recuperado el 10 de 10 de 2015, de: http://www.eltiempo.com/colombia/otras-ciudades/robo-de-motos-en-colombia-aumentan-los-casos-en-el-2015/15831737

El universal. (2012). Se incrementó robo de motos y atracos de fleteros en Sahagún. Recuperado el 21 de 11 de 2013, de http://www.eluniversal.com.co/monteria-y-sincelejo/sucesos/se-incremento-robo-de-motos-y-atracos-de-fleteros-en-sahagun-66089

Enrique Amodeo. (2010). Servicios web (2): ¿Qué es REST?. Recuperado el 03 de 11 de 2015, de https://eamodeorubio.wordpress.com/2010/07/26/servicios-web-2-%C2%BFque-es-rest/

Francisco Águila V. (2012). Emol.com. Gobierno lanza nuevo plan para evitar robo de motocicletas en el país. Recuperado el 22 de 09 de 2014, de http://www.emol.com/noticias/nacional/2012/05/31/543300/gobierno-lanza-nuevo-plan-para-evitar-robo-de-motocicletas-en-el-pais.html

IBM. (2015). IBM Knowledge Center. Qualities of service provided by an MQTT client. Recuperado el 07 de 11 de 2015, de http://www-01.ibm.com/support/knowledgecenter/SSFKSJ_7.5.0/com.ibm.mq.dev.doc/q029090_.htm?lang=es

José Antonio Yébenes Gálvez. (2015). GeekyTheory. ¿Qué es MQTT?. Recuperado el 07 de

11 de 2015, de https://geekytheory.com/que-es-mqtt/

José Luis Cano. (2007). Elmundo.es¡. Adiós a los robos de motos!. Recuperado el 21 de 11 de 2013, de http://www.elmundo.es/elmundomotor/2007/05/24/usuarios/1180021269.html

JosmanTek. (2014). Restful APIs – ¿Que es REST?. Recuperado el 03 de 11 de 2015, de http://blog.josmantek.com/php/restful-apis-que-es-rest/

Juan Antonio de la Puente. (2007). Introducción a los sistemas de tiempo real.

Julián Rodríguez Fernández. (2013). Circuito cerrado de televisión y seguridad electrónica.

Kevin Ashton. (2009). That 'Internet of Things' Thing.

Laura Verdeguer Asins. (2008). Elmundo.es. Detector presenta un dispositivo para localizar motos robadas. Recuperado el 21 de 11 de 2013, de http://www.elmundo.es/elmundomotor/2008/07/28/usuarios/1217259560.html

Luis Miguel Gracia. (2013). Un poco de java. Un poco de MQTT. Recuperado el 07 de 11 de 2015, de https://unpocodejava.wordpress.com/2013/02/27/un-poco-de-mqtt/

Luis Miguel Gracia. (2012). Un poco de java. ¿Qué es MQTT?. Recuperado el 07 de 11 de 2015, de https://unpocodejava.wordpress.com/2012/12/06/que-es-mqtt/

Microsoft Corporation. (2006). La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real.

Iván Prieto. (2015). Internet of Things en la práctica; Manos a la obra.
Mis Respuestas. (2009). ¿Qué es un servidor web. Recuperado el 25 de 05 de 2014, de http://www.misrespuestas.com/que-es-un-servidor-web.html

MongDB. (2014). Recuperado el 27 de 05 de 2014, de http://www.mongodb.org/

Motor. (2015). Colombia es un país que devora motocicletas. Recuperado el 30 de 04 de 2015, de http://www.motor.com.co/carros-motor/colombia-un-pais-que-devora-motos_15207956-4

Motor. (2015). Se dispararon las ventas de motos en Colombia durante el 2014. Recuperado el 30 de 04 de 2015, de http://www.motor.com.co/carros-motor/disparadas-las-ventas-de-motos-en-2014_15074839-4

Motor Pasión Moto. (2013). Un sistema de seguridad para mi moto, ¿Quizás Detector

OnBike?. Recuperado el 25 de 04 de 2015, de http://www.motorpasionmoto.com/tecnologia/un-sistema-de-seguridad-para-mi-moto-quizas-detector-onbike

Nodejs. (2014). Recuperado el 27 de 05 de 2014, de http://nodejs.org/

Oscar Vega Pava. (2012). La Motocicleta Como Medio De Transporte. Recuperado el 21 de 11 de 2013, de http://www.redjbm.com/catedra/index.php/gente/19-la-motocicleta-como-medio-de-transporte

Portafolio. (2012). Portafolio.co. Tecnología GPS reduciría robos de motocicletas en Colombia. Recuperado el 21 de 11 de 2013, de http://www.portafolio.co/negocios/gps-reduciria-robos-motocicletas

Rafael Enríquez Herrador. (2009). Guía de Usuario de Arduino. Recuperado el 21 de 11 2013 de: http://www.uco.es/aulasoftwarelibre/wpcontent/uploads/2010/05/Arduino_user_manual_es.pdf

Roberto Ruiz. (SF). Motos.about.com. Tipos de antirrobos de moto. Recuperado el 13 de 11 de 2013, de http://motos.about.com/od/accesorios-moto/tp/Tipos-De-Antirrobos-De-Moto.htm

Secure Bike (2013). Proyecto Sistema De Seguridad Para Moto. Recuperado el 21 de 11 de 2013, de http://www.youtube.com/watch?v=E5s22Gc0yms

SOFTENG. (2015). Metodologia Scrum para desarrollo de software – aplicaciones complejas. Recuperado el 09 de 11 de 2015, de https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html

Tecnología de los Plásticos. (2011). Recuperado el 21 de 11 de 2013 de http://tecnologiadelosplasticos.blogspot.com/2011/06/baquelita.html

Theodore Hope. (2009). Sistemas Escalables

Thunder. (2012). Alarma de alejamiento Thunder. Recuperado el 21 de 11 de 2013, de http://www.youtube.com/watch?v=VjVhSzqtJAU

Vanguardia. (2013). Por seguridad, usted podrá marcar su moto en Bucaramanga. Recuperado el 13 de 11 de 2013, de http://www.vanguardia.com/santander/bucaramanga/216067-por-seguridad-usted-podra-marcar-su-moto-en-bucaramanga

Vanguardia. (2010). Operativos para prevenir el robo de motocicletas en Santander. Recuperado el 21 de 11 de 2013, de http://www.vanguardia.com/historico/67228-operativos-

para-prevenir-el-robo-de-motocicletas

Wikipedia, La enciclopedia libre. (2015, 21 de septiembre). Arquitectura orientada a servicios. Fecha de consulta: octubre 30, 2015 desde https://es.wikipedia.org/w/index.php?title=Arquitectura_orientada_a_servicios&oldid=85261425.

Wikipedia, La enciclopedia libre. (2014, 17 de mayo). Sistema de posicionamiento global. Fecha de consulta: mayo 26, 2014 desde http://es.wikipedia.org/w/index.php?title=Sistema_de_posicionamiento_global&oldid=74461652

Wikipedia, La enciclopedia libre. (2014, 20 de marzo). Aplicación móvil. Fecha de consulta: mayo 26, 2014 desde http://es.wikipedia.org/w/index.php?title=Aplicaci%C3%B3n_m%C3%B3vil&oldid=7 3303035.

Wikipedia, La enciclopedia libre. (2015, 30 de octubre). Códigos de estado HTTP. Fecha de consulta: noviembre 03, 2015 desde https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP

Wikipedia, La enciclopedia libre. (2014, 9 de mayo). Dispositivo móvil. Fecha de consulta: mayo 26, 2014 desde http://es.wikipedia.org/w/index.php?title=Dispositivo_m%C3%B3vil&oldid=7429267 9.

Wikipedia, La enciclopedia libre. (2014, 2 de mayo). Interfaz de usuario. Fecha de consulta: mayo 16, 2014 desde http://es.wikipedia.org/w/index.php?title=Interfaz_de_usuario&oldid=74160828.

Wikipedia, La enciclopedia libre. (2014, 7 de junio). Navegador web. Fecha de consulta: mayo 25, 2014 desde http://es.wikipedia.org/w/index.php?title=Navegador_web&oldid=74901297.

Wikipedia. (2013).Botón (dispositivo). Fecha de consulta: 21 de 11 de 2014 de http://es.wikipedia.org/wiki/Bot%C3%B3n_(dispositivo)

Wikipedia. (2013). Relé o Relevador. Fecha de consulta: 21 de 11 de 2014 desde http://es.wikipedia.org/wiki/Rel%C3%A9

ANEXOS

Anexos 1. Cronograma De Actividades

CRONOGRAMA DE ACTIVIDADES																
ACTIVIDAD \ MES	AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
	SEM 1	SEM 2	SEM 3	SEM 4	SEM 1	SEM 2	SEM 3	SEM 4	SEM 1	SEM 2	SEM 3	SEM 4	SEM 1	SEM 2	SEM 3	SEM 4
Estudio de las tecnología a utilizar																
Diseño de la infraestructura del servidor																
Diseño de la infraestructura del sitio web																
Desarrollo del servidor																
Desarrollo del sitio web																
Diseño y desarrollo de la aplicación móvil (Android)																
Desarrollo del prototipo que va dentro de la moto(Arduino)																
Pruebas unitarias de cada componente de la plataforma y sus respectivas correcciones																
Pruebas y correcciones de todo el sistema integrado																
Presentación de todo el sistema funcionando																

Figura 50: Cronograma de actividades

Anexos 2. Manual de instalación y despliegue del servidor de la API y la aplicación web.

A continuación se detallaran los pasos a seguir para tener la aplicación web y la API funcionando correctamente en Heroku y obteniendo datos de MongoLab.

Este manual especifica como instalar y ejecutar correctamente la aplicación utilizando el sistema operativo Ubuntu versión 15.10.

Estos pasos también pueden ser utilizados para ejecutar esto en cualquier otra distribución de Linux o cualquier otro sistema operativo, pero teniendo en cuenta que el proceso de instalación de algunos componentes puede variar.

Para esto se deben seguir los siguientes pasos:

1. **Crear una cuenta en MongoLab.**
2. **Iniciar sesión en MongoLab.**
3. **Crear la base de datos en MongoLab.**
4. **Crear un usuario con los permisos para acceder a dicha base de datos en MongoLab.**
5. **Crear una cuenta en Heroku.**
6. **Instalar Heroku Toolbelt en nuestro computador.**
7. **Obtener el código fuente del Servidor web y API.**
8. **Instalar Git en nuestro computador.**
9. **Iniciar un repositorio Git en la carpeta con el código fuente del servidor.**
10. **Iniciar sesión en Heroku desde nuestra terminal.**
11. **Crear una aplicación en Heroku desde nuestra terminal.**
12. **Agregar archivos, hacer commit con Git.**
13. **Hacer push con Git hacia Heroku.**
14. **Verificar la aplicación en el Dashboard de Heroku.**
15. **Abrir la página web.**

1. **Crear una cuenta en MongoLab.**

   Para esto nos dirigimos a https://mongolab.com/signup/ y llenamos el formulario de registro.

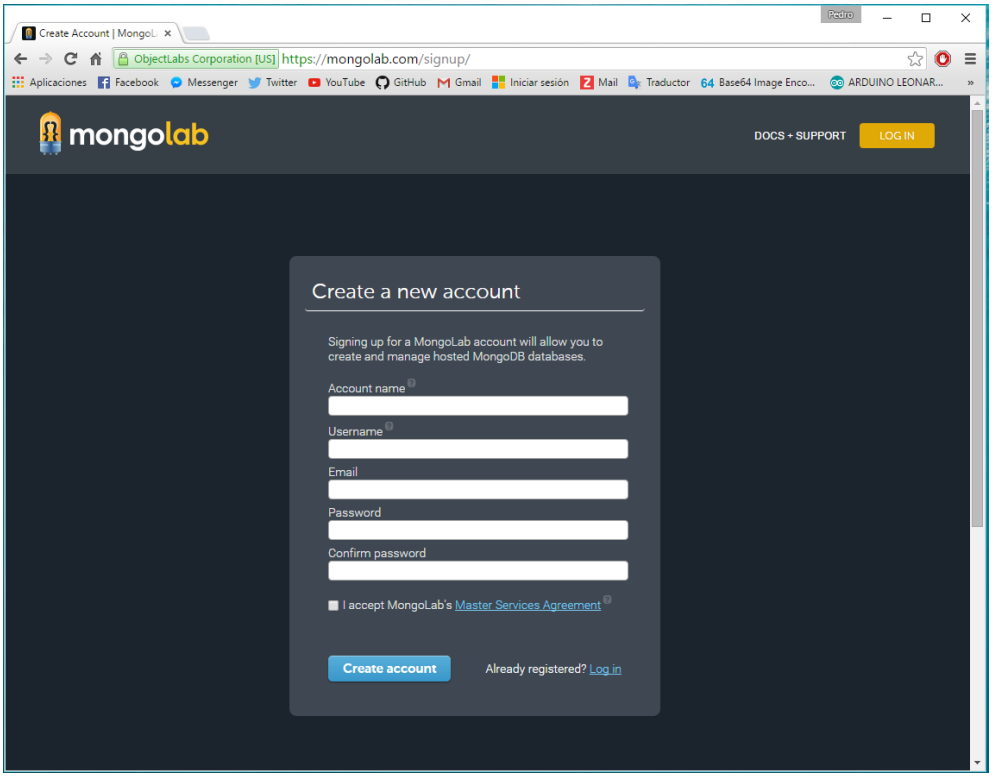

Figura 51: Creación de cuenta de usuario en MongoLab.com

2. **Iniciar sesión en MongoLab.**

   Después de haber creado nuestra cuenta nos dirigimos a https://mongolab.com/login/ para iniciar sesión.

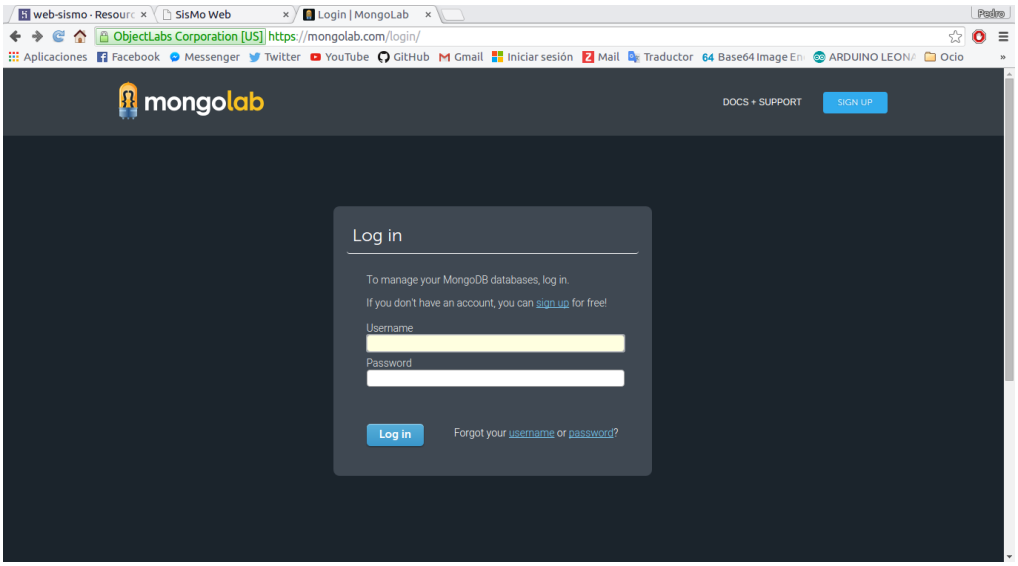

Figura 52: Inicio de sesión en MongoLab.com

3. **Crear la base de datos en MongoLab.**

Después de iniciar sesión, damos click en un botón con un rayo amarillo que dice Create new, aparecerá una ventana en donde piden nombre de la base de datos algunos otros datos más.

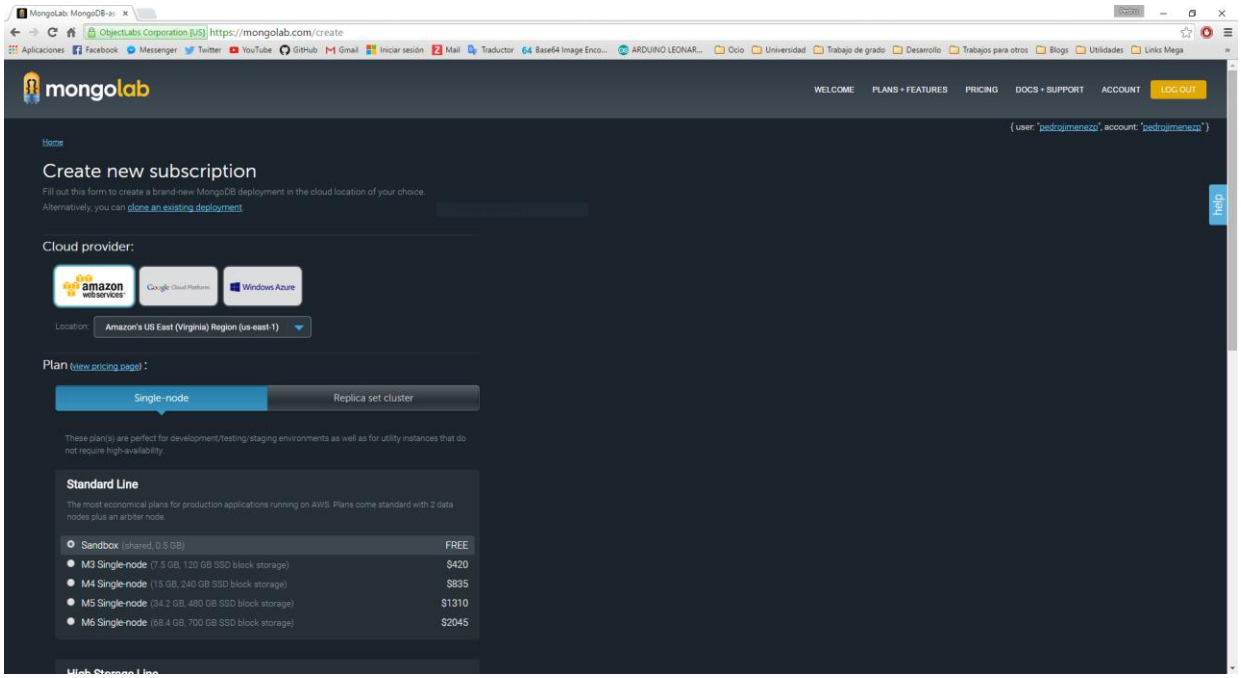

Figura 53: Ventana de creación de una nueva base de datos

Al dar click en el botón créate se crea una nueva base de datos como se puede observar en la siguiente figura.

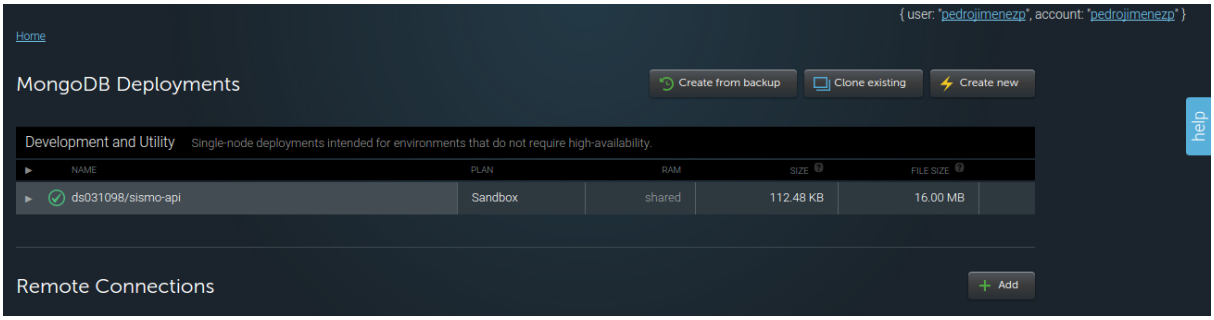

Figura 54: Base de datos creada

4. **Crear un usuario con los permisos para acceder a dicha base de datos en MongoLab.**

Al crear la base de datos damos click en la fila de la tabla y se nos aparecerá la siguiente página:

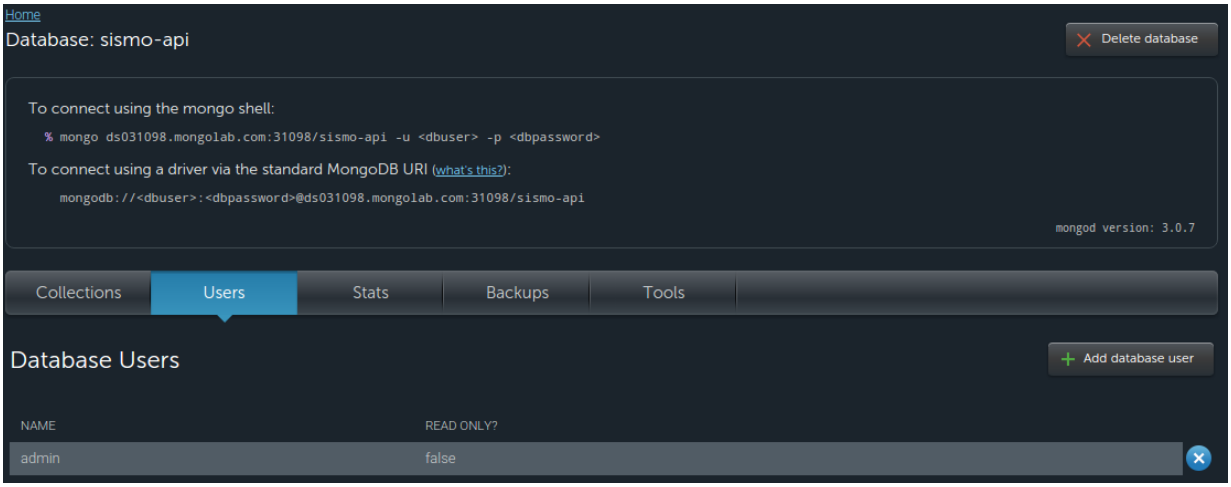

Figura 55: Vista detallada de la base de datos

En la figura 55 se puede observar que en la parte de arriba se muestra la manera de conectarse a esta base de datos tanto desde la consola de nuestro computador como desde el código fuente de nuestro servidor usando un driver para mongodb.

Para crear un usuario para esta base de datos damos click en el botón **Add database user** y aparecerá una ventana en donde ingresamos el nombre de usuario y la contraseña.

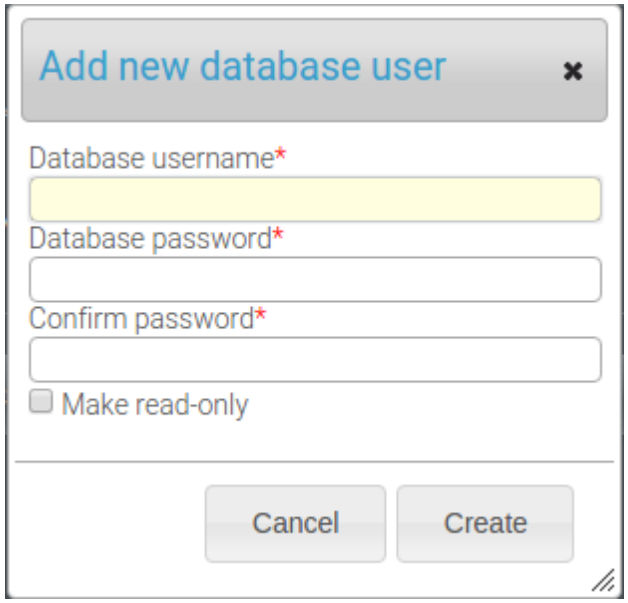

Figura 56: Creación del usuario de la base de datos

Al dar click en el botón créate nos aparecerá un usuario en el apartado **Users**.

5. Crear una cuenta en Heroku.

Para crear una cuenta en Heroku nos dirigimos a https://signup.heroku.com/ y llenamos el formulario.

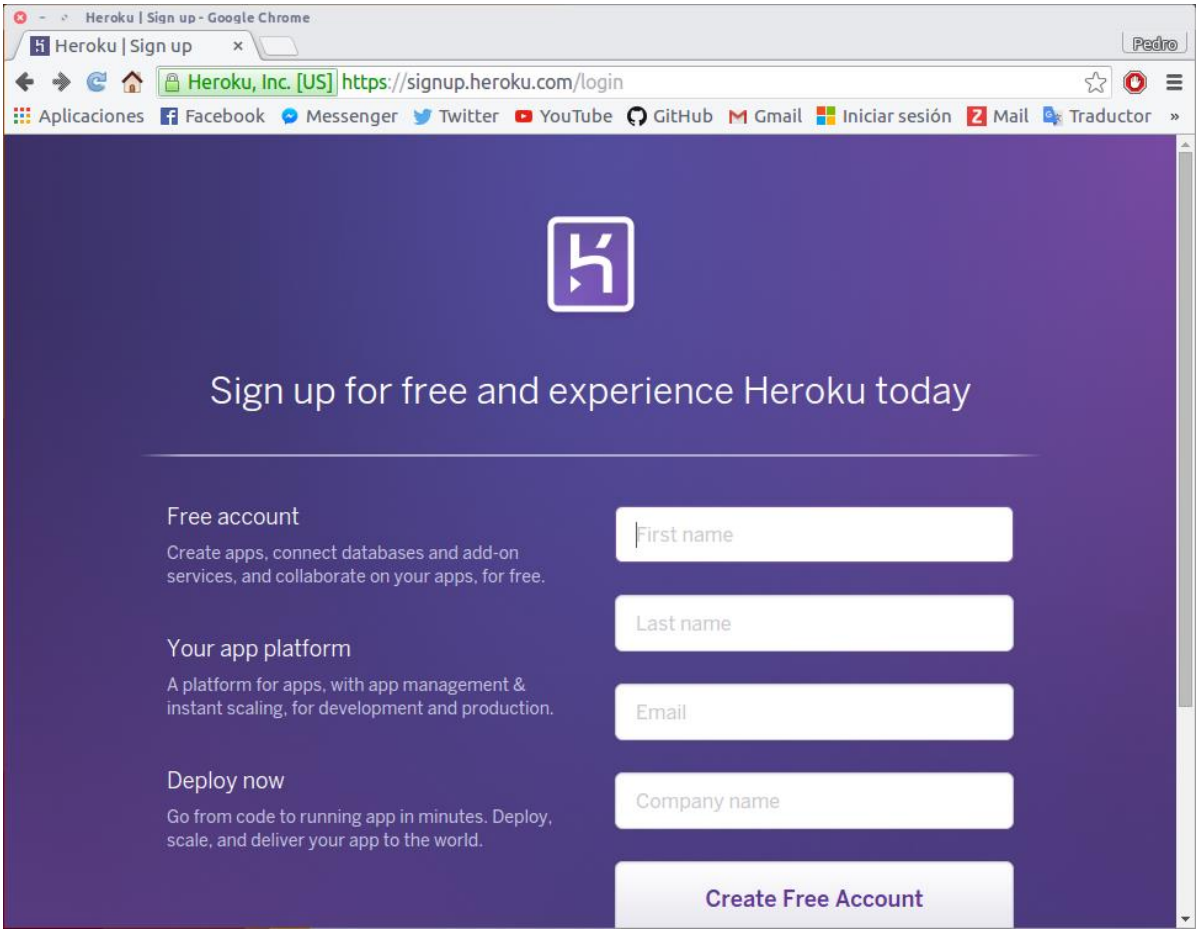

Figura 57: Registro en Heroku.com

6. **Instalar Heroku Toolbelt en nuestro computador.**

   Para instalar Heroku Toolbelt diríjase a https://devcenter.heroku.com/articles/getting-started-with-nodejs#set-up y descargue el instalador para su sistema operativo.

7. **Obtener el código fuente del Servidor web y API.**

   El código fuente del Servidor web y API está en una misma carpeta llamada **sismo_web_server** que será entregada con este documento.

8. **Instalar Git en nuestro computador.**

Para instalar Git en nuestro computador abrimos una terminal y escribimos lo siguiente: **sudo apt-get install git**.

```
pedro@dell-laptop ~ sudo apt-get install git
[sudo] password for pedro:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
git ya está en su versión más reciente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
pedro@dell-laptop ~
```

Figura 58: Instalación de git en Ubuntu 15.10

9. **Iniciar un repositorio Git en la carpeta con el código fuente del servidor.**

   Para esto nos dirigimos desde la terminal a la carpeta en donde tengamos el código fuente que previamente hemos obtenido y escribimos: **git init**

```
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master cd
pedro@dell-laptop ~ cd GitHub/sismo-platform/sismo-web-server
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master ls -l
total 32
drwxrwxr-x  3 pedro pedro 4096 nov 12 07:43 build
drwxr-xr-x  4 pedro pedro 4096 nov 11 20:47 client
-rw-rw-r--  1 pedro pedro 1081 ago 24 10:41 LICENSE
drwxrwxr-x 16 pedro pedro 4096 nov 12 07:41 node_modules
-rw-rw-r--  1 pedro pedro  802 nov 12 08:33 package.json
-rw-r--r--  1 pedro pedro   28 nov 12 07:34 Procfile
-rw-rw-r--  1 pedro pedro   53 ago 24 10:41 README.md
drwxrwxr-x  3 pedro pedro 4096 ago 24 10:41 src
```

Figura 59: Cambiarse a la carpeta del código fuente desde la consola

.

```
reset
✘ pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server git init
```

Figura 60: Ejecutando git int

10. **Iniciar sesión en Heroku desde nuestra terminal.**

    Para ejecutar sesión en Heroku desde la terminal simplemente escribimos: **Heroku login** y después escribimos el nombre de usuario y contraseña con la que creamos la cuenta anteriormente.

```
pedro@dell-laptop ~ cd GitHub/sismo-platform/sismo-web-server
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master heroku login
Enter your Heroku credentials.
Email: pedroj.jimenezp@gmail.com
Password (typing will be hidden):
Logged in as pedroj.jimenezp@gmail.com
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master
```

Figura 61: Iniciando sesión en Heroku desde la terminal

11. Crear una aplicación en Heroku desde nuestra terminal.

Después de haber iniciado sesión en Heroku desde la terminal, verificamos que estamos en el directorio raíz de la carpeta del código fuente y procedemos a escribir el siguiente comando: **heroku apps:create web-sismo**

```
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master heroku apps:create
web-sismo
Creating web-sismo... done, stack is cedar-14
https://web-sismo.herokuapp.com/ | https://git.heroku.com/web-sismo.git
Git remote heroku added
```

Figura 62: Creación de la aplicación web en Heroku desde la terminal

Esto creara una aplicación con el nombre **web-sismo** en Heroku.com y además agregara el repositorio remoto a Git para hacer deploy fácil simplemente usando **git push heroku master.**

12. Agregar archivos, hacer commit con Git.

Después de crear la aplicación en Heroku con **heroku create** procedemos a agregar desde la consola los archivos a git y hacer un commit.

Para esto escribimos lo siguiente en la terminal: **git add . && git commit –m "commit"**

```
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master git add .
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master + git commit -m "c
ommit"
[master (root-commit) 314ebc9] commit
 4434 files changed, 627699 insertions(+)
 create mode 100644 LICENSE
 create mode 100644 Procfile
```

Figura 63: Agregando archivos a git y haciendo commit

13. Hacer push con Git hacia Heroku.

Después de agregar los archivos a git y hacer commit, simplemente resta enviar todo el código a Heroku.com haciendo uso de git utilizando el siguiente comando: **git push heroku master**

```
pedro@dell-laptop ~/GitHub/sismo-platform/sismo-web-server master git push heroku ma
ster
Counting objects: 4139, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3671/3671), done.
Writing objects: 100% (4139/4139), 5.63 MiB | 100.00 KiB/s, done.
Total 4139 (delta 508), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:        NPM_CONFIG_LOGLEVEL=error
remote:        NPM_CONFIG_PRODUCTION=true
remote:        NODE_ENV=production
remote:        NODE_MODULES_CACHE=true
```

Figura 64: Haciendo push a Heroku.com a través de git

```
remote: -----> Compressing... done, 17.3MB
remote: -----> Launching... done, v3
remote:        https://web-sismo.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy.... done.
To https://git.heroku.com/web-sismo.git
 * [new branch]      master -> master
```

Figura 65: Confirmación del push hacia Heroku.com

14. Verificar la aplicación en el Dashboard de Heroku.

Después de haber realizado el push hacia Heroku.com nos vamos a su página, iniciamos sesión y verificamos que el nombre de nuestra aplicación este entre la lista de aplicaciones.

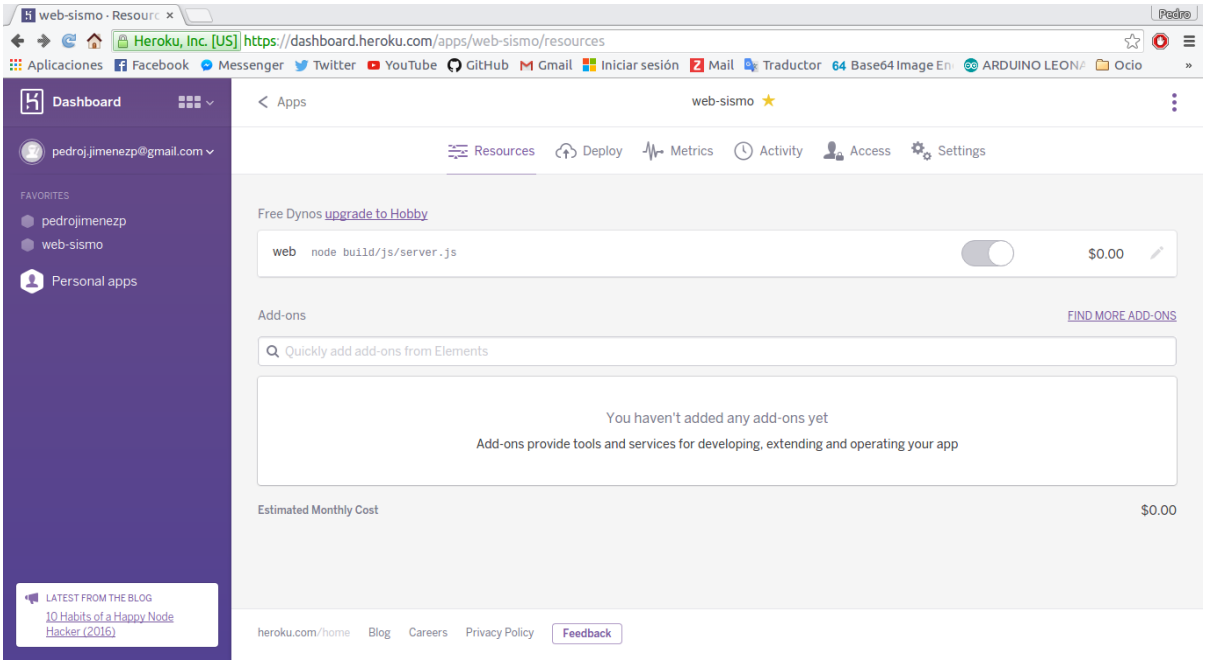

Figura 66: Verificando que la aplicación este en el dashboard de Heroku.com

15. Abrir la página web.

Si la aplicación se encuentre en el dashboard y todo se ha realizado correctamente, nos dirigimos a la url con este formato https://[nombre_de_la_app].herokuapp.com en nuestro caso sería https://web-sismo.herokuapp.com y nos debe mostrar el sitio web como se muestra a continuación.

Figura 67: Aplicación web de SISMO

Anexos 3. Manual de uso de la aplicación móvil.

En este manual se describirá información clara y concisa de cómo utilizar la aplicación móvil del sistema de seguridad para motocicletas **SisMo**.

SisMo App es una aplicación móvil que puede ser accedida desde cualquier móvil con sistema operativo Android desde la versión 4.1 en adelante.

Acceso a la aplicación

Para acceder a la aplicación, el usuario deberá buscar el icono de la App en su móvil y proceder a abrirla:

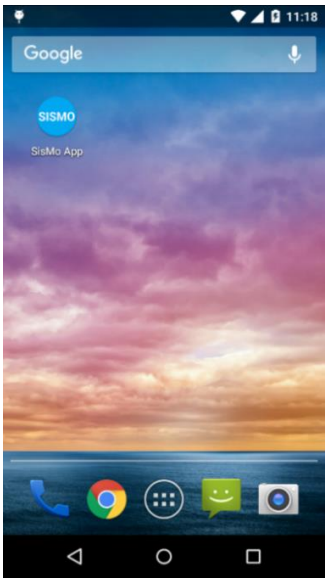

Figura 68: Icono de la aplicación móvil

Una vez abierta la aplicación, se visualiza una pantalla en donde se le solicita al usuario ingresar los datos de autenticación:

Figura 69: Pantalla de inicio de sesión

Si se trata de un usuario no registrado, este deberá presionar el botón “REGISTRARSE” que se encuentra ubicado en la parte inferior de la pantalla “Iniciar sesión”. La anterior acción desplegará la pantalla “Registrarse” en donde se le solicita al usuario ingresar los datos para el registro en la App. El usuario deberá llenar el formulario y presionar el botón “REGISTARSE” de dicha pantalla.

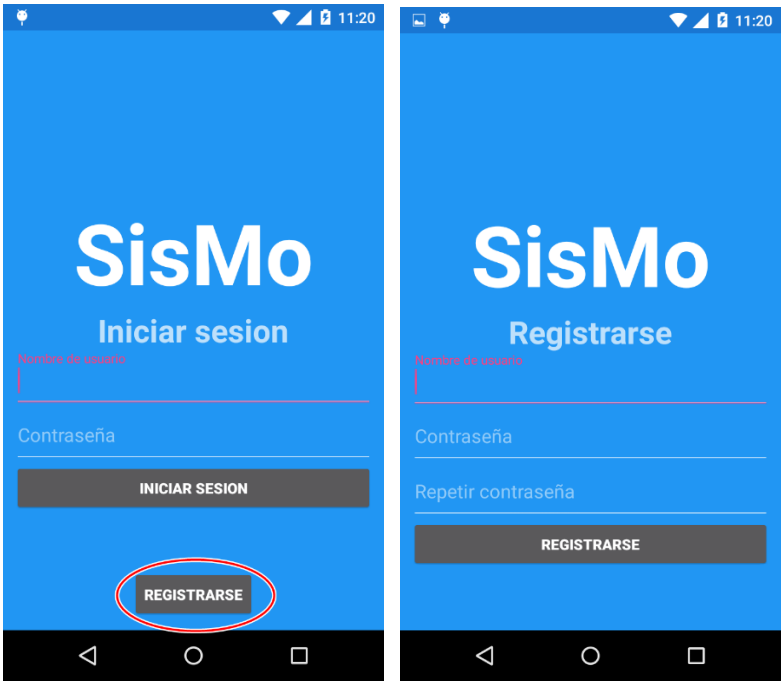

Figura 70: Botón "REGISTRARSE" y pantalla "Registrarse"

Una vez el usuario se ha registrado, automáticamente entra a la pantalla “Home” y puede comenzar a utilizar la aplicación.

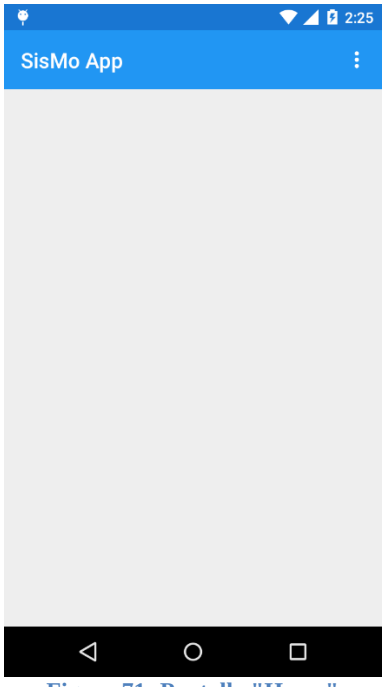

Figura 71: Pantalla "Home"

Si el usuario ya tiene una cuenta en la App, simplemente debe ingresar en la pantalla “Iniciar sesión”, su nombre de usuario, su contraseña y presionar el botón “INICIAR SESION”.

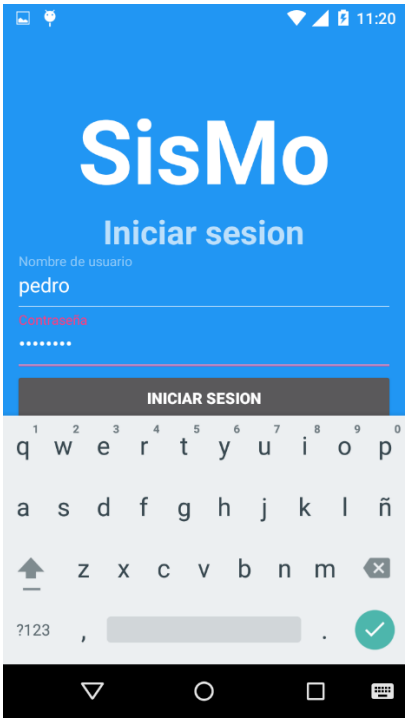

Figura 72: Usuario ingresando datos de autenticación

Si los datos ingresados por el usuario no son correctos el sistema visualiza un mensaje de error “Usuario o contraseña incorrectos”, como se observa a continuación:

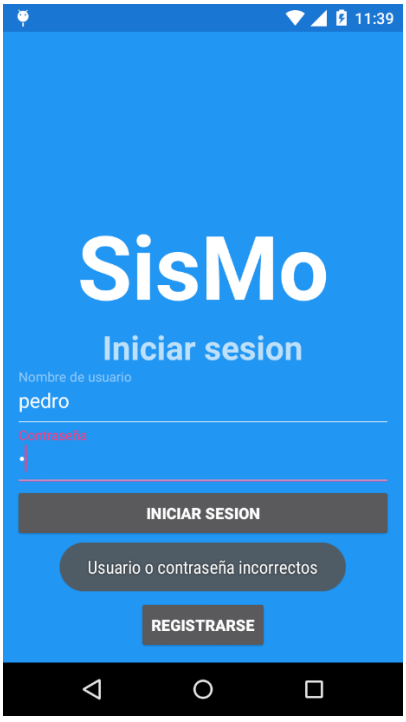

Figura 73: Mensaje de error en la autenticación

Si el nombre de usuario y la contraseña ingresada son válidos, el sistema le ofrece al usuario las opciones a las que tiene privilegios.

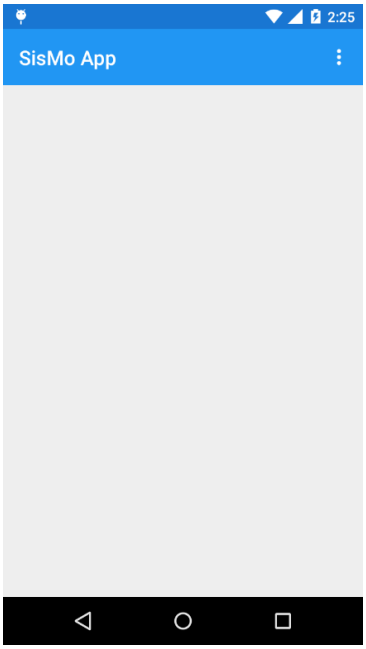

Figura 74: Autenticación correcta "Home"

SisMo ofrece los siguientes privilegios (funcionalidades) a los usuarios que inicien sesión en la aplicación:

1. Registrar Motocicletas.
2. Ver motocicletas.
3. Actualizar datos de una motocicleta.
4. Eliminar motocicleta.
5. Mostrar estado de los elementos monitoreados (mostrar posición de moto, mostrar estado del monitoreo, mostrar estado del seguro, mostrar estado del fluido eléctrico).
6. Activar monitoreo (notificar estado del seguro, notificar desplazamientos, rastrear motocicleta).
7. Reportar robo.
8. Desactivar monitoreo.
9. Cerrar sesión.
10. Notificaciones de la App.
11. Rastrear motocicleta.

A continuación veremos la guía de uso para cada una de las funciones anteriores del sistema.

1. Registrar motocicletas

Para registrar una motocicleta el usuario debe dirigirse a la parte superior izquierda de la pantalla “Home” y desplegar el menú ubicado en este punto. En la siguiente imagen dicho menú se encuentra señalado con un círculo rojo para hacer más fácil su identificación:

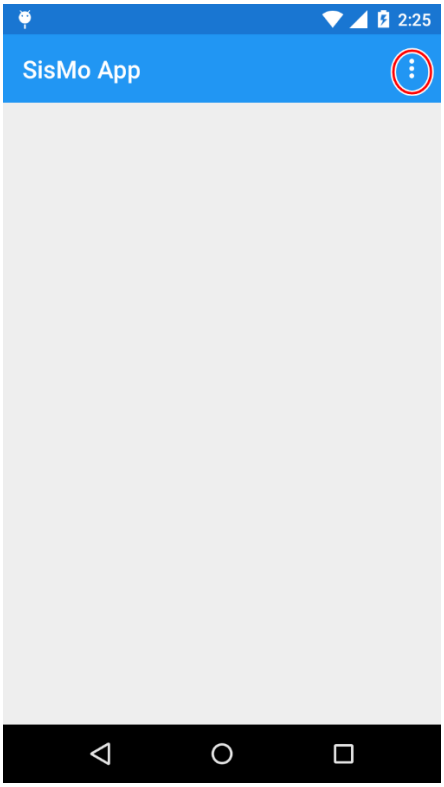

Figura 75: Menú desplegable

Se selecciona la primera opción de la lista que se despliega del menú: “Agregar moto”, e inmediatamente se visualiza la pantalla donde se le solicita al usuario ingresar la imagen y los datos de su vehículo:

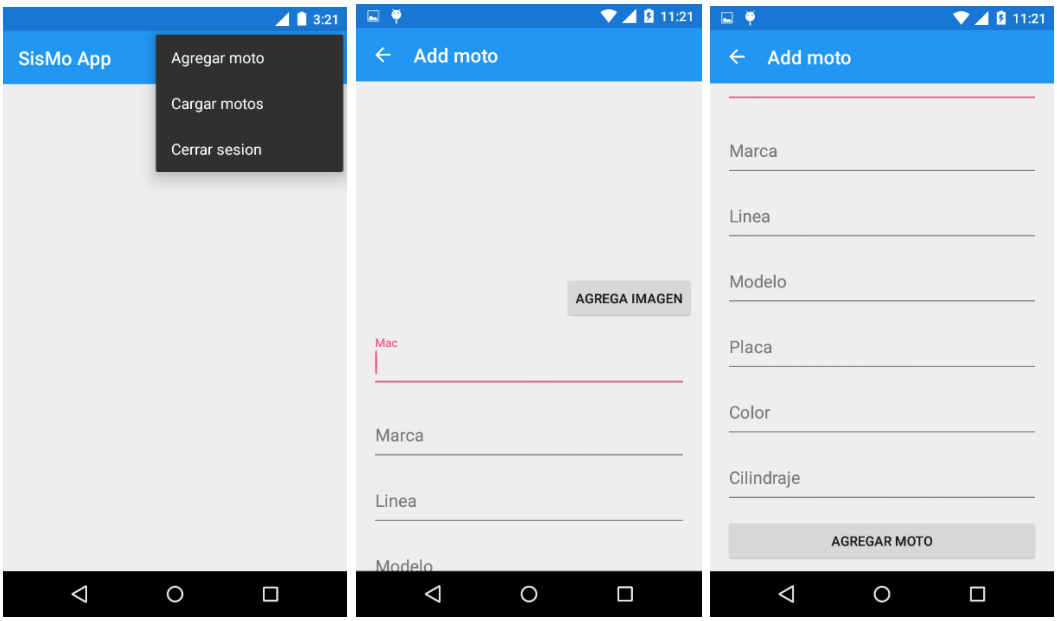

Figura 76: Opción "AGREGAR MOTO" y pantalla "Agregar moto"

Una vez que el usuario haya llenado el formulario de registro de su vehículo debe presionar el botón “AGREGAR MOTO” para lograr un registro exitoso. Después de esto se visualizará una pantalla con la foto y los datos de la motocicleta registrada. El usuario podrá agregar la

cantidad de motos que desee.

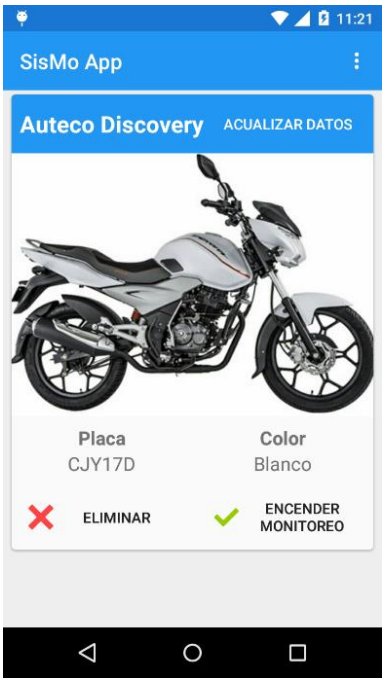

Figura 77: Lista de motocicletas registradas

2. Ver motocicletas

Para ver las motocicletas registradas basta con que el usuario entre a la aplicación, donde se visualizará una pantalla con todas sus motocicletas registradas.

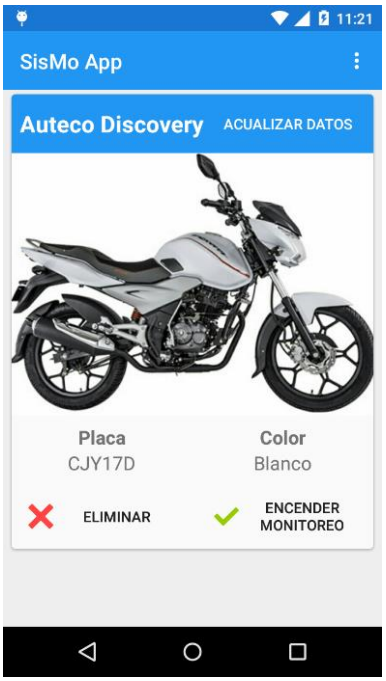

Figura 78. Lista de motocicletas registradas

El usuario también puede utilizar la segunda opción del menú desplegable “Cargar motos”, para refrescar la lista de sus motocicletas.

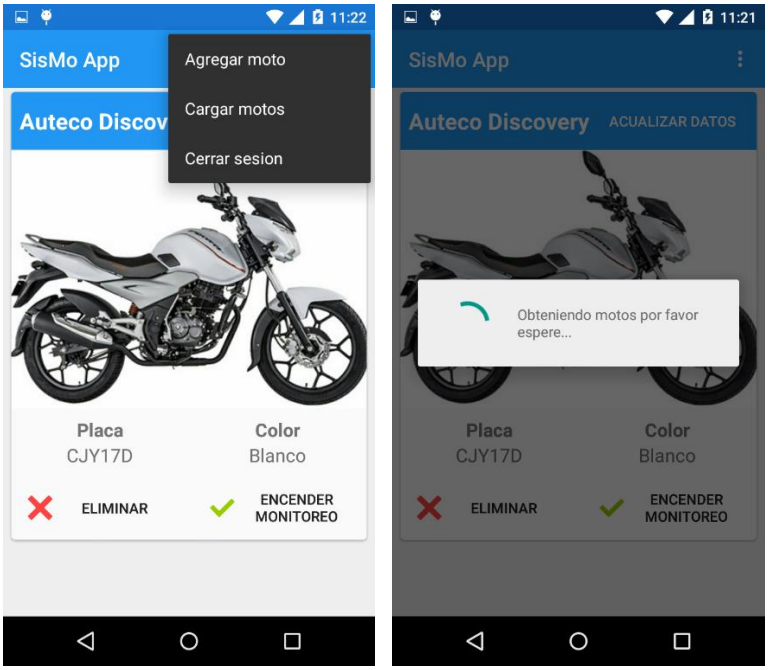

Figura 79: Opción "Cargar motos"

3. Actualizar datos de una motocicleta

Para actualizar los datos de una motocicleta el usuario debe ir al listado de sus motocicletas (**Ver ítem 2**)**,** ubicar la motocicleta a la que desee actualizar sus datos y en la parte superior izquierda de la vista de su moto presionar el botón “ACTUALIZAR DATOS”:

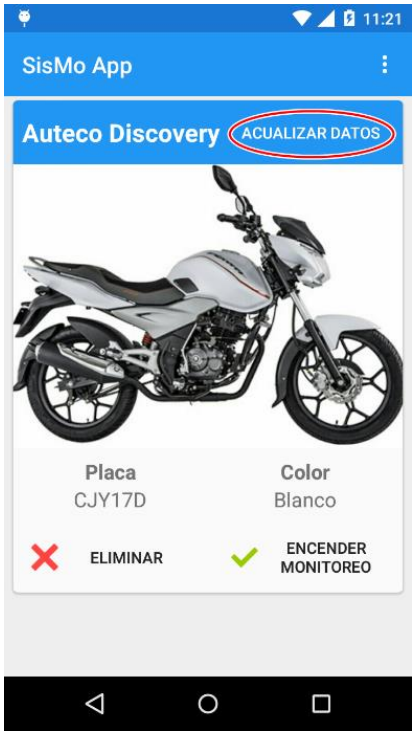

Figura 80: Botón "ACTUALIZAR DATOS"

Después de presionar en la opción señalada con el círculo rojo, se visualizará el formulario para actualizar los datos de la motocicleta. El usuario debe digitar la información que pretenda actualizar de su motocicleta y presionar el botón “ACTUALIZAR INFORMACION”:

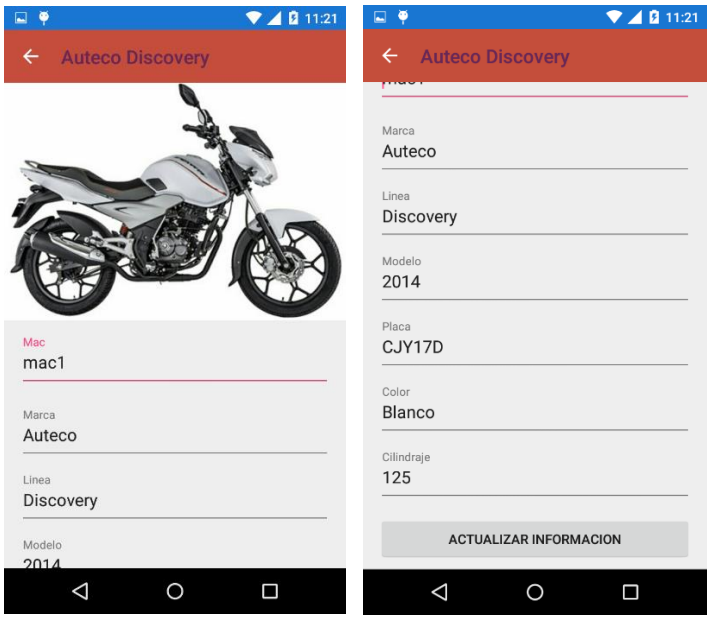

Figura 81: Formulario y botón para actualizar motocicleta

4. Eliminar motocicleta

Para eliminar una motocicleta el usuario debe ir al listado de sus motocicletas (**Ver ítem 2**), ubicar la motocicleta que desee eliminar y en la parte inferior derecha de la vista de su moto el botón “ELIMINAR”:

Figura 82: Botón "ELIMINAR"

La anterior acción lanza el siguiente mensaje de confirmación:

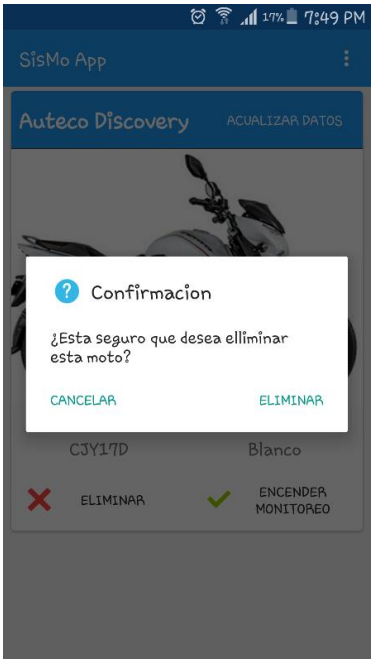

Figura 83: Mensaje de confirmación para eliminar moto

Al seleccionar “CANCELAR” no se elimina la motocicleta, pero si el usuario elige “ELIMINAR”, inmediatamente la motocicleta se borra de la lista.

5. Mostrar estado de los elementos monitoreados (estado del monitoreo, estado del seguro, estado del fluido eléctrico, mostrar posición de la motocicleta)

Para ver el estado de los elementos monitoreados de una motocicleta, el usuario debe ir al listado de sus motocicletas (**Ver ítem 2**), una vez allí debe ubicar la motocicleta a la que desee ver el estado de sus elementos y presionar sobre ella para que se despliegue una nueva pantalla con el estado de cada uno de los elementos monitoreados:

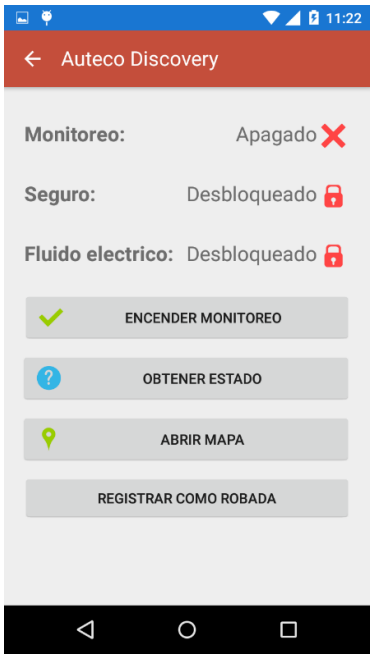

Figura 84: Pantalla de elementos monitoreados

Si el usuario quiere verificar si ha sucedido algún cambio en el estado de dichos elementos aun sin activar el monitoreo, debe presionar el botón “OBTENER ESTADO”, para actualizar el estado de cada uno de los elementos monitoreados

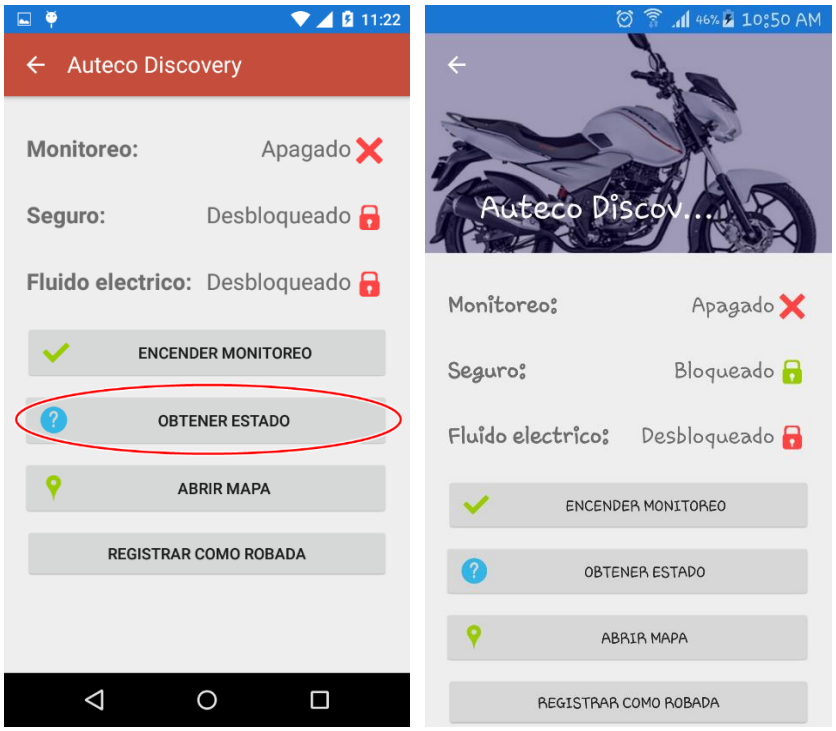

Figura 85: Botón "OBTENER ESTADO"

Si el usuario quiere conocer la posición actual de su motocicleta, debe presionar el botón “ABRIR MAPA” como se señala a continuación:

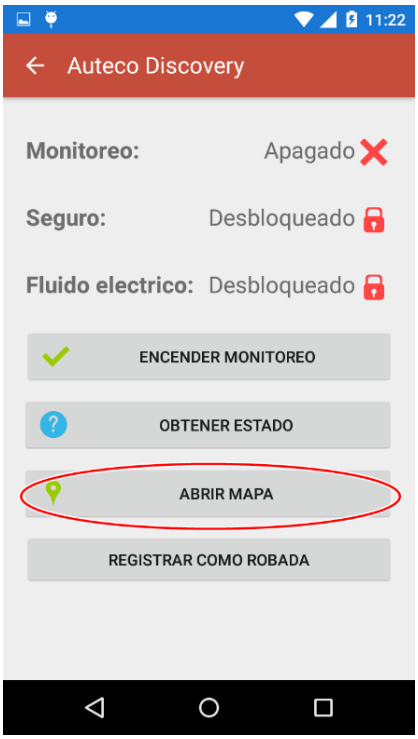

Figura 86: Botón "ABRIR MAPA"

La anterior acción lanzará un mensaje de selección donde el usuario deberá elegir la opción “1) Abrir el mapa y pedir la posición de la moto.” Como se puede ver a continuación:

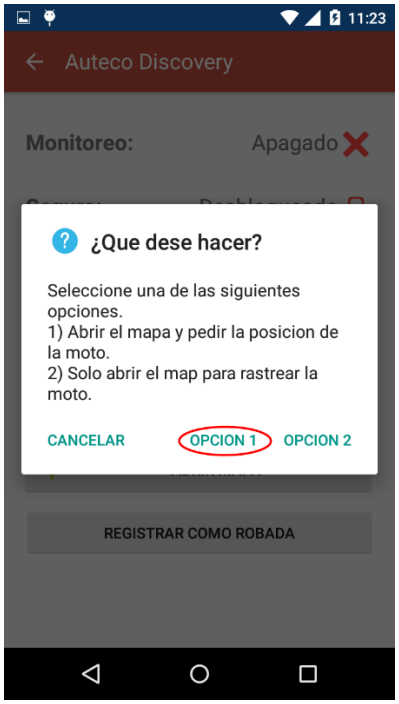

Figura 87: Mensaje de selección para abrir mapa

Al seleccionar la “OPCION 1” inmediatamente se mostrará en pantalla el mapa de Google Maps con dos puntos de referencia:

Rojo: Posición actual de la motocicleta
Azul: Posición del usuario

Figura 88: Mapa de Google Maps

6. Activar monitoreo

Activar el monitoreo se puede hacer de dos formas distintas:

Desde la vista general de motocicletas registradas.

El usuario debe ir al listado de sus motocicletas (**Ver ítem 2**), una vez allí debe ubicar la motocicleta a la que le desee activar el monitoreo y presionar sobre el botón “ENCENDER MONITOREO” como se señala a continuación:

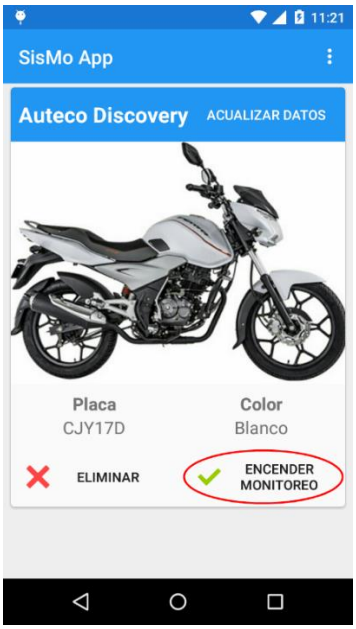

Figura 89: Botón "ENCENDER MONITOREO"

Al ejecutar la anterior acción se mostrará en pantalla el siguiente mensaje de confirmación:

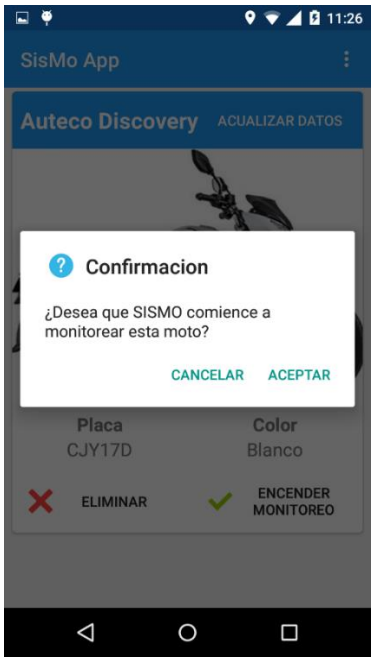

Figura 90: Mensaje de confirmación para activar monitoreo

Si el usuario quiere activar el monitoreo deberá escoger la opción “ACEPTAR”, en caso contrario, deberá escoger la opción “CANCELAR”.

Se puede verificar que el monitoreo está activo pues la parte superior de la vista de la motocicleta cambia de color (de azul a verde) y el botón “ENCENDER MONITOREO” pasa a “APAGAR MONITOREO” como se nota a continuación:

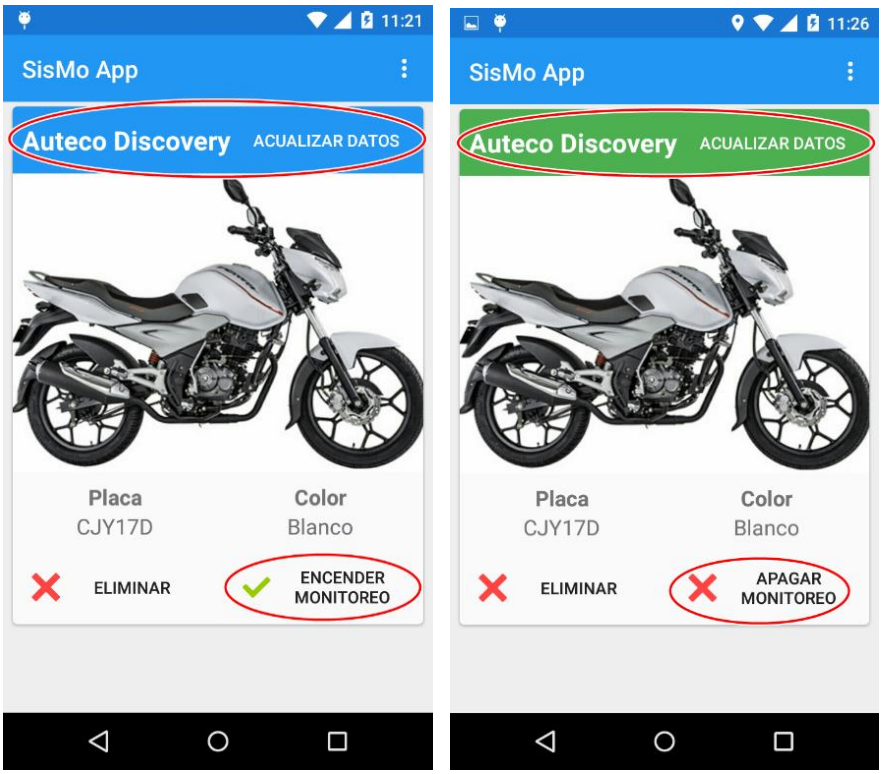

Figura 91: Verificación de monitoreo activado

Desde la vista de una motocicleta especifica:

El usuario debe ir al listado de sus motocicletas (**Ver ítem 2**), una vez allí debe ubicar la motocicleta a la que le desee activar el monitoreo y presionar sobre ella para que se despliegue la pantalla de elementos monitoreados. Luego deberá presionar el primer botón, “ENCENDER MONITOREO”:

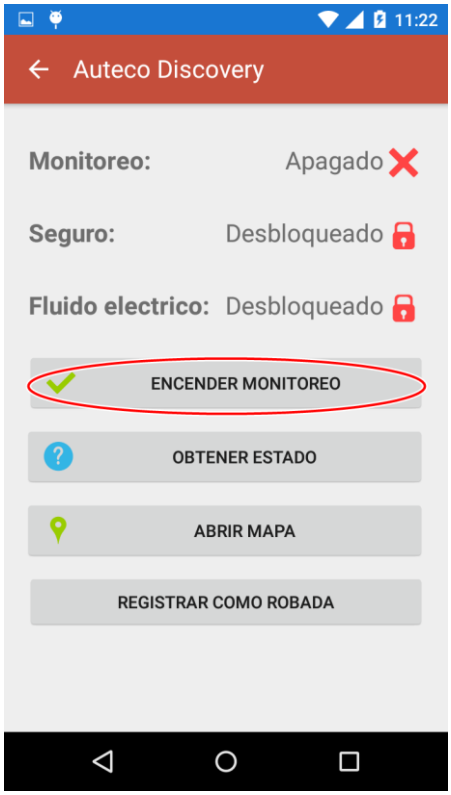

Figura 92: Encender monitoreo segunda forma

Al ejecutar la anterior acción se mostrará en pantalla el siguiente mensaje de confirmación:

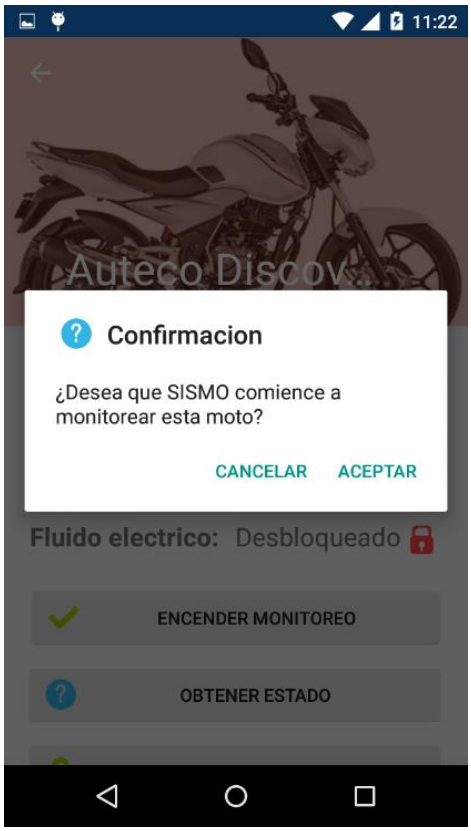

Figura 93: Mensaje de confirmación para activar monitoreo segunda forma

Si el usuario quiere activar el monitoreo deberá escoger la opción “ACEPTAR”, en caso contrario, deberá escoger la opción “CANCELAR”.

Se puede verificar que el monitoreo está activado cuando el botón “ENCENDER MONITOREO” pasa a “APAGAR MONITOREO” como se nota a continuación:

Figura 94: verificación de monitoreo activado segunda forma

Al activar el monitoreo la aplicación se encargará de mostrarle al usuario notificaciones automáticas sobre cambios en el estado del seguro, cambios en la posición de la motocicleta y en caso de robo permitir rastrear al vehículo. Las anteriores notificaciones se muestran al final de este documento.

7. Registrar robo

Para registrar una motocicleta como robada, el usuario debe ir al listado de sus motocicletas (**Ver ítem 2**), una vez allí debe ubicar la motocicleta que desea registrar como robada y presionar sobre ella para que se despliegue la pantalla de elementos monitoreados, donde aparece el botón “REGISTRAR COMO ROBADA”:

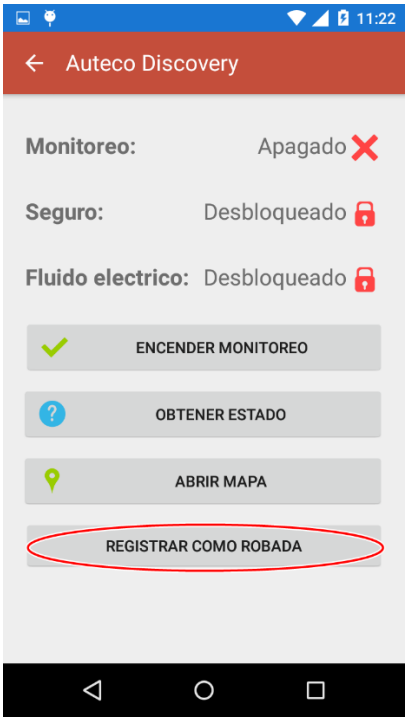

Figura 95: Botón "REGISTRAR COMO ROBADA"

Al presionar dicho botón, aparecerá un mensaje de confirmación, donde el usuario deberá escoger la opción “REGISTRAR” para registrar exitosamente una motocicleta como robada.

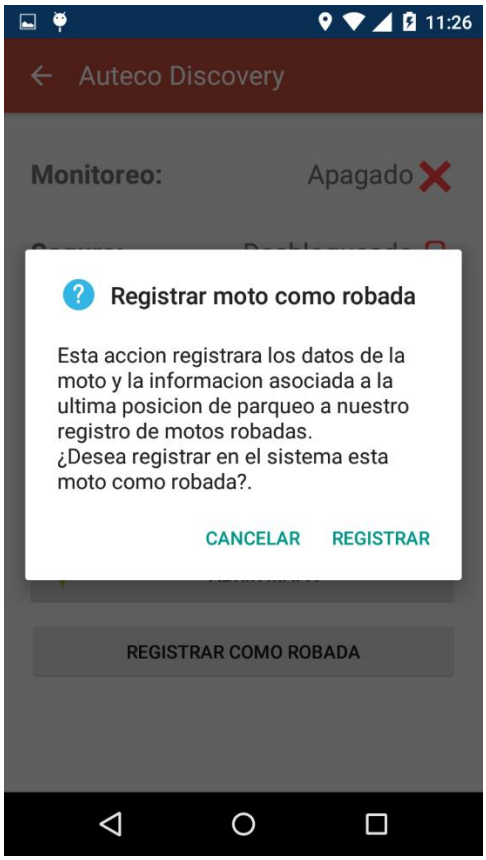

Figura 96: Mensaje de confirmación para registrar moto como robada

El reporte de la motocicleta robada deberá aparecer en la página web del sistema.

8. Desactivar monitoreo

Para desactivar el monitoreo, simplemente se hacen los mismos pasos que cuando se va a activar el monitoreo, teniendo en cuenta que cambia el nombre del botón a: “APAGAR MONITOREO”.

9. Cerrar sesión

Para cerrar la sesión, el usuario debe dirigirse al menú desplegable y escoger la última opción “Cerrar sesión” como se observa en la siguiente imagen:

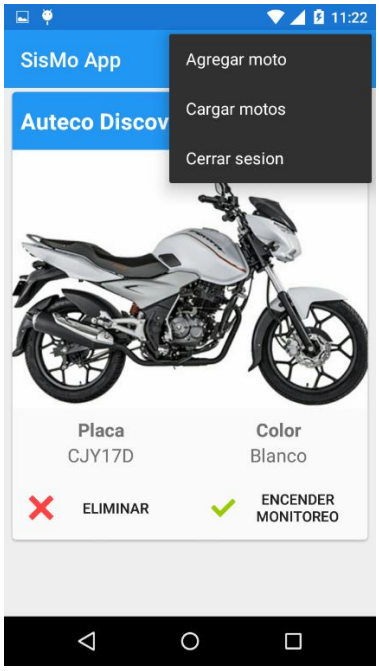

Figura 97: Cerrar sesión

10. Notificaciones de la App:

Las notificaciones se muestran de forma diferente dependiendo de los siguientes estados de la aplicación.

Aplicación cerrada: Cuando la aplicación se encuentra cerrada las notificaciones se mostraran en el móvil del usuario como se observa a continuación:

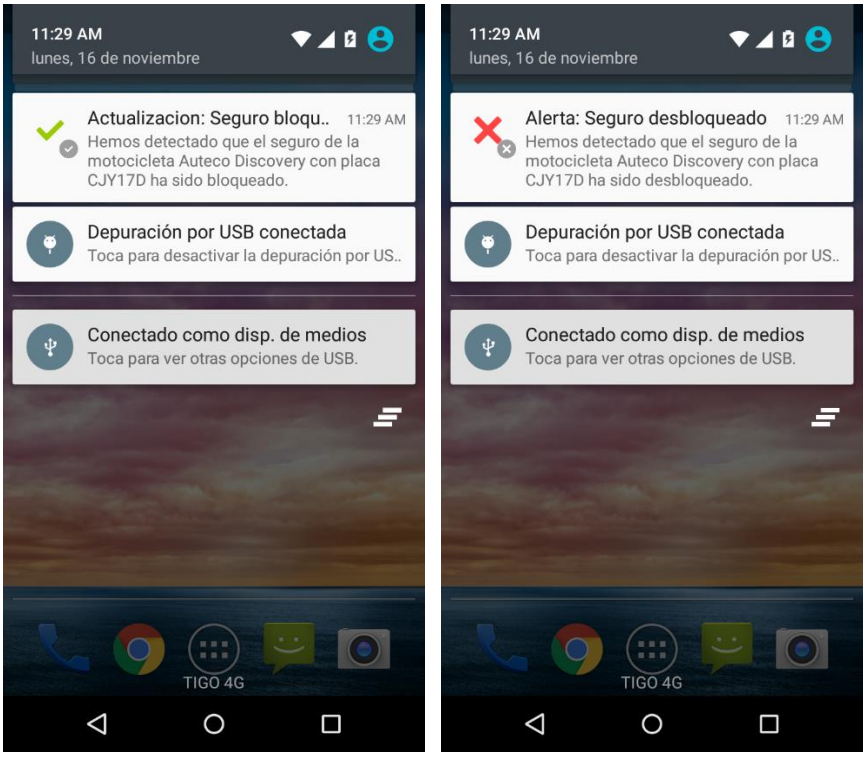

Figura 98: Notificaciones sobre cambios en el estado del seguro

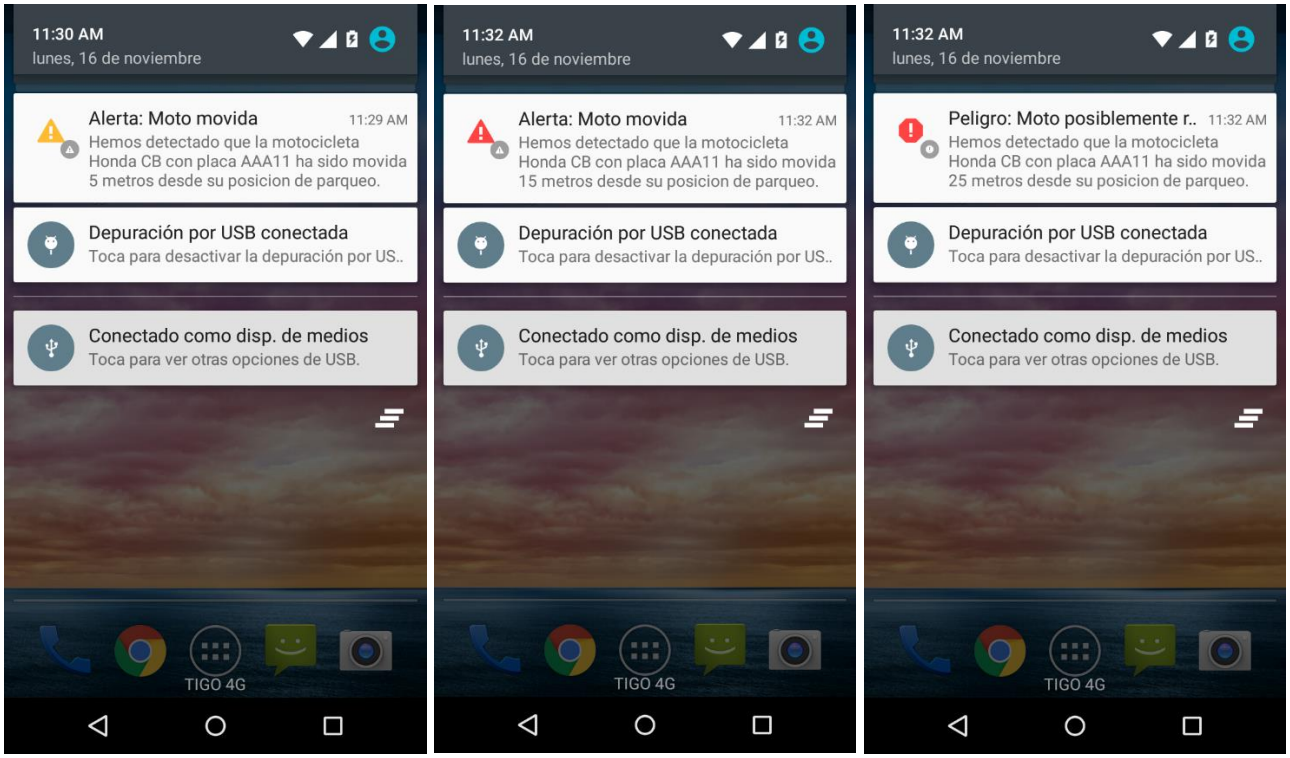

Figura 99: Notificaciones de desplazamiento de motocicleta a los 5, 15 y 25 metros

Aplicación abierta: cuando la aplicación se encuentre abierta se mostraran las notificaciones de dos formas dependiendo en que vista se encuentre el usuario:

Desde la vista general de motocicletas registradas.

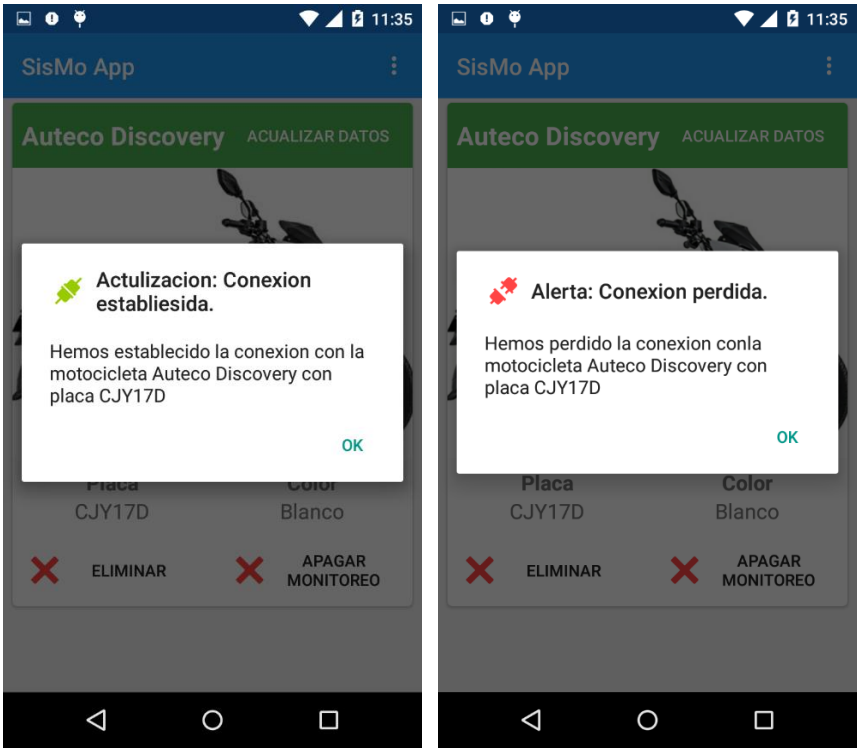

Figura 100: Notificación del estado de la conexión

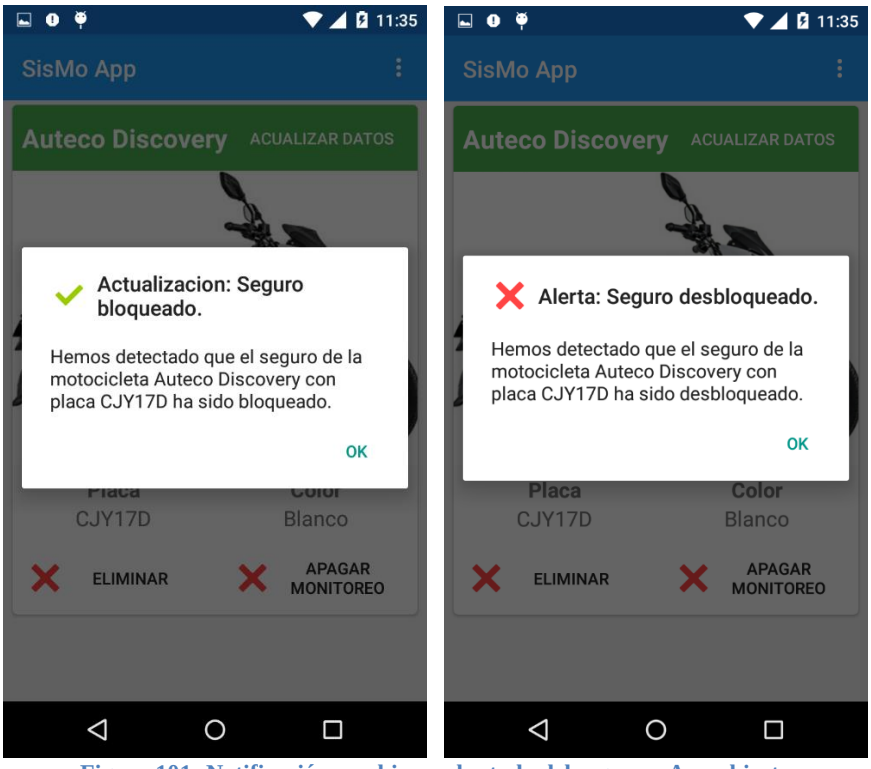

Figura 101: Notificación cambio en el estado del seguro – App abierta

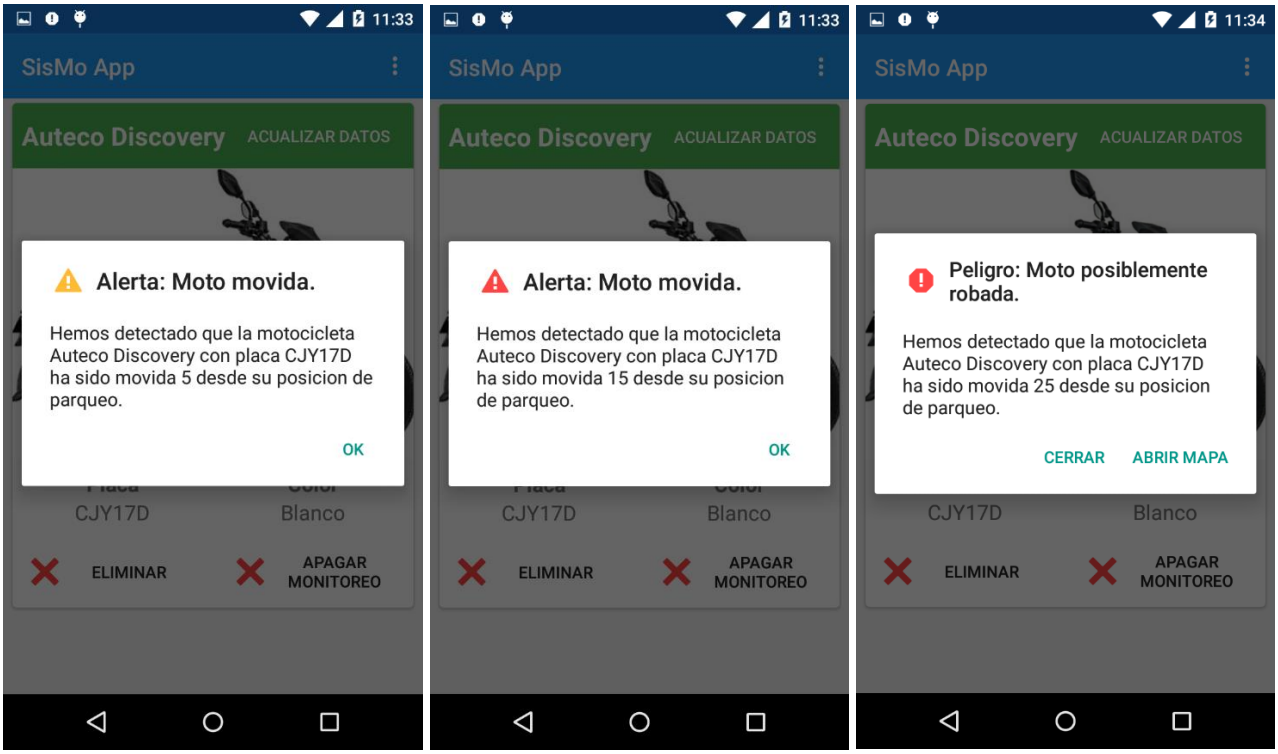

Figura 102: Notificaciones de desplazamiento de motocicleta a los 5, 15 y 25 metros - App abierta

Desde la vista de una motocicleta especifica:

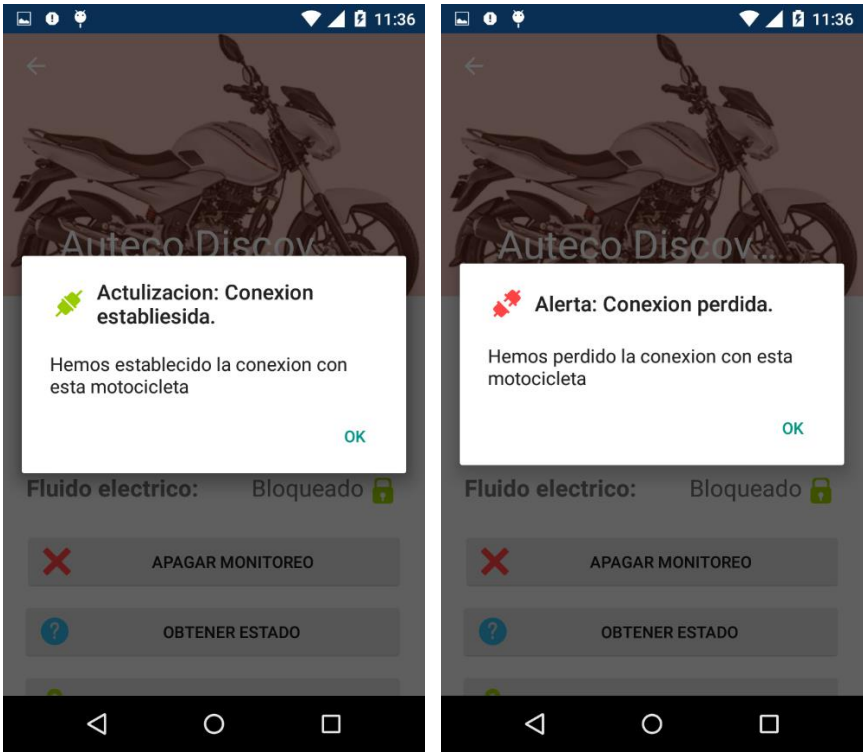

Figura 103: Notificación del estado de la conexión - Vista especifica

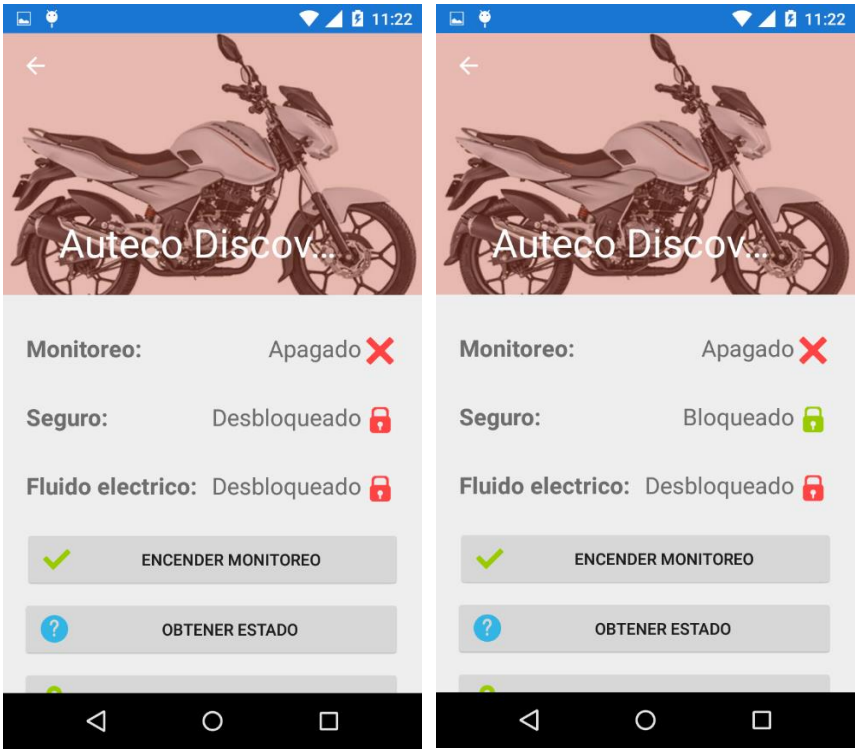

Figura 104: Estado de los elementos monitoreados con el monitoreo apagado

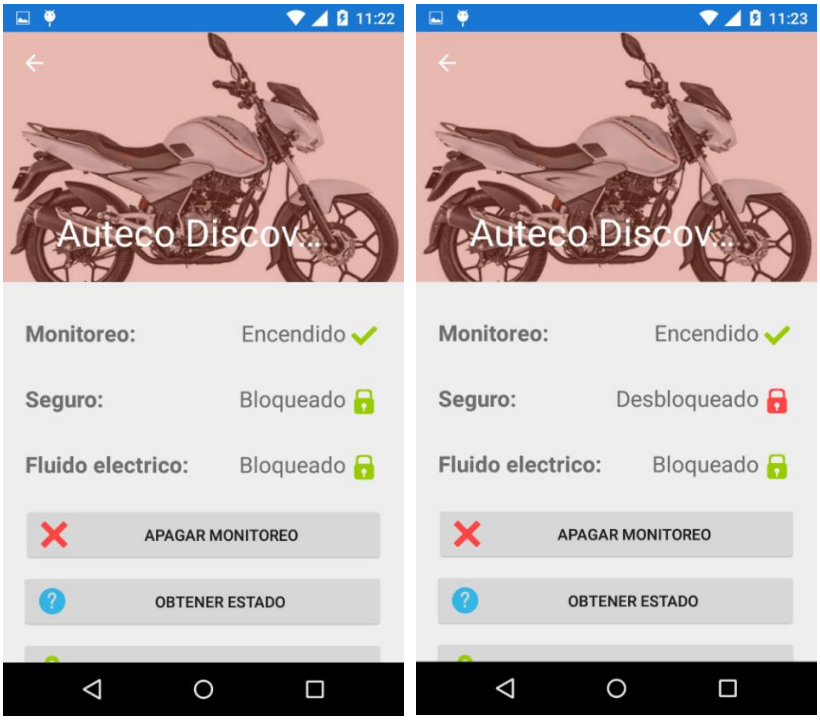

Figura 105: estado de los elementos monitoreados con el monitoreo activado

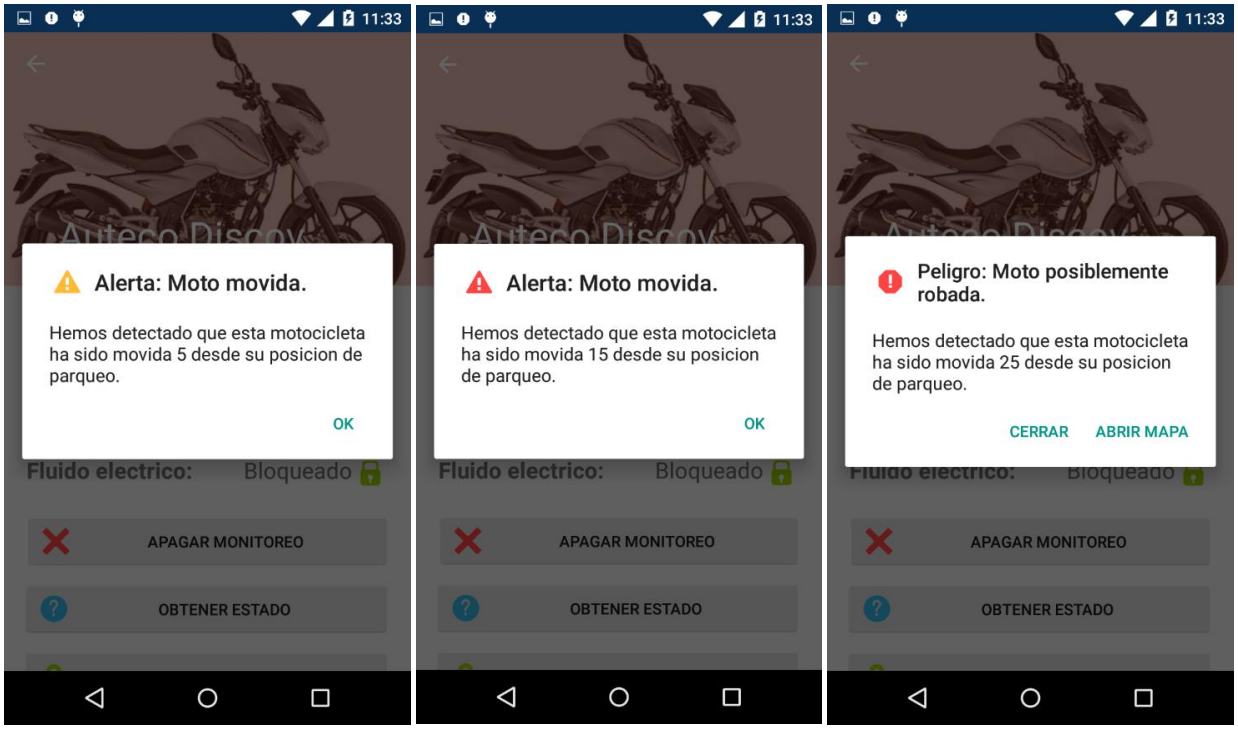

Figura 106: Notificaciones de desplazamiento de motocicleta a los 5, 15 y 25 metros - App abierta - Vista especifica

11. Rastrear motocicleta:

Cuando el usuario tenga el monitoreo activado y su motocicleta sea movida 25 metros de distancia de su posición inicial de parqueo, inmediatamente se mostrará en pantalla la notificación correspondiente a este hecho, mostrando la opción de “ABRIR MAPA” que permite rastrear la motocicleta en tiempo real por medio de Google Maps:

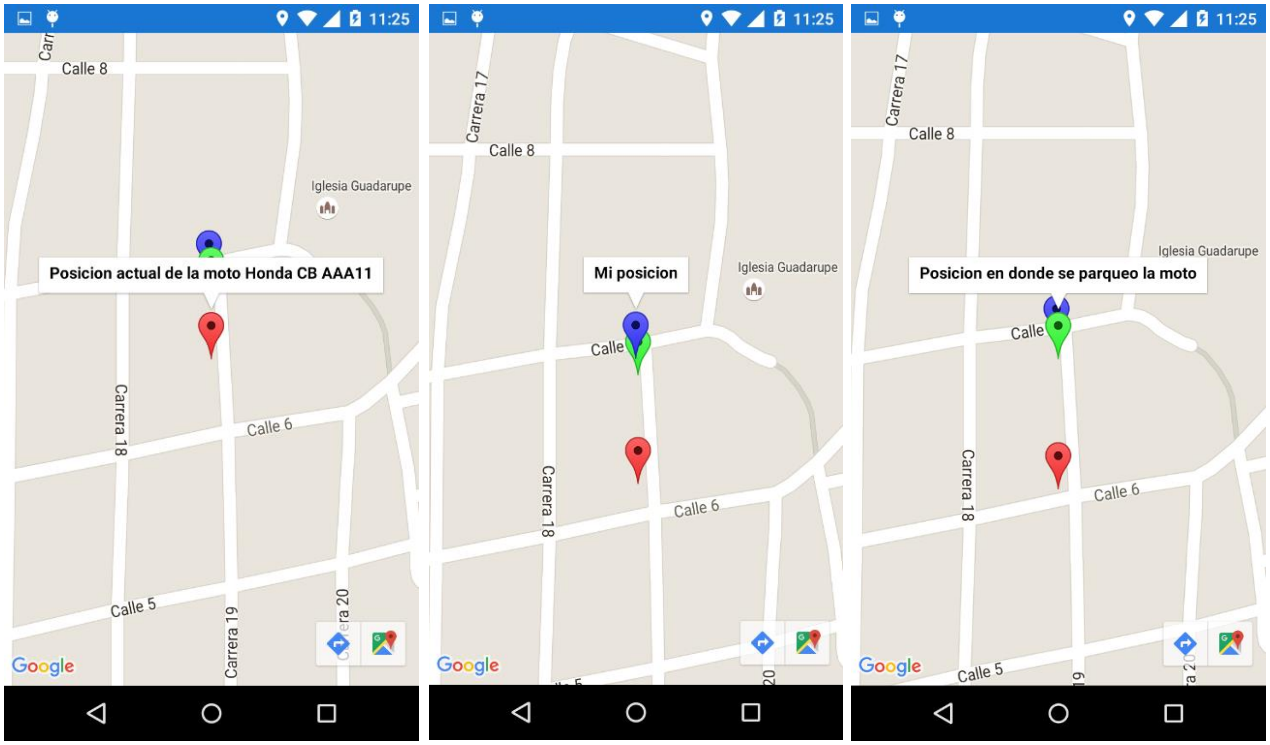

Figura 107: Rastreo de motocicleta

Como se puede observar, en el mapa anterior se muestran tres puntos de referencia:

Azul: Posición actual del usuario (Dispositivo móvil del usuario).
Verde: Posición donde el usuario dejó parqueada la motocicleta.
Rojo: Posición por donde va la motocicleta.